Clustering with Constraints

Sugato Basu

SRI Intl. (AI Center)

Email: basu@ai.sri.com

Ian Davidson

SUNY – Albany (Dept. of Computer Science)

Email: davidson@cs.albany.edu

Acknowledgements

- Contribution of slides
  - Tomer Hertz
  - Sepandar Kamvar
  - Brian Kulis
- Insightful discussions, comments
  - James Bailey
  - S.S. Ravi
  - Kiri Wagstaff
- Apologies
  - If we do not get around to covering your work or if you have work on constraints and clustering and we didn’t include it in the bibliography (drop us an email).

Notation

- S : set of training data
- s_i : i^{th} point in the training set
- L: cluster labels on S
- l_i : cluster label of s_i
- C_j: centroid of j^{th} cluster
- ML : set of must-link constraints
- CL : set of cannot-link constraints
- CC_i : a connected component (sub-graph)
- TC : the transitive closure
- D(x,y) : Distance between two points x and y

Outline

- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
    - Enforcing constraints [Ian]
    - Hierarchical [Ian]
    - Learning distances [Sugato]
    - Initializing and pre-processing [Sugato]
    - Graph-based [Sugato]

Outline

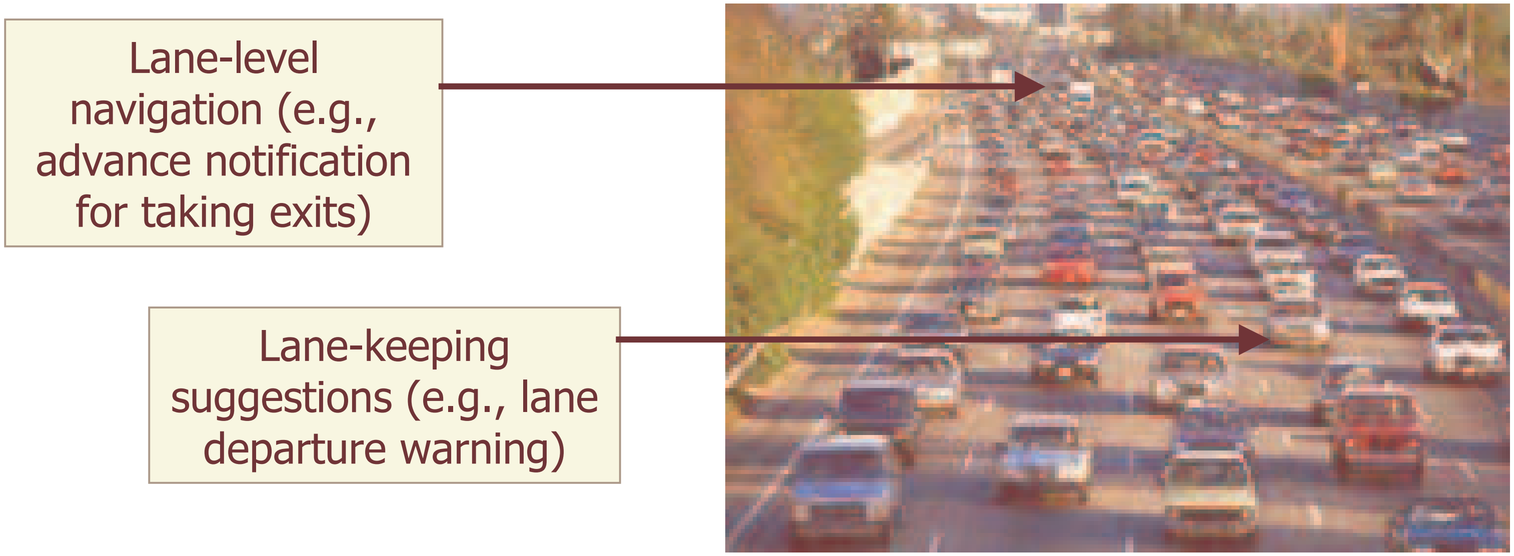
- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
    - Enforcing constraints [Ian]
    - Hierarchical [Ian]
    - Learning distances [Sugato]
    - Initializing and pre-processing [Sugato]
    - Graph-based [Sugato]

Motivating Examples in Non-Hierarchical Clustering

- Given a set of instances *S*
- Find the "best" set partition

  $S = \{S_1 \cup S_2 \cup \ldots S_k\}$
- Multitude of algorithms that define "best" differently
  - K-Means
  - Mixture Models
  - Self Organized Maps
- Aim is to find novel and actionable patterns …

Automatic Lane Finding from GPS traces [Wagstaff et al. ’01]



Lane-level navigation (e.g., advance notification for taking exits)

Lane-keeping suggestions (e.g., lane departure warning)

- **Constraints inferred from trace-contiguity (ML) & max-separation (CL)**

Mining GPS Traces (Schroedl et' al)

- Instances are represented by the *x*, *y* location on the road. We also know when a car changes lane, but not what lane to.
- Desired clusters are very elongated, horizontally aligned central lines.

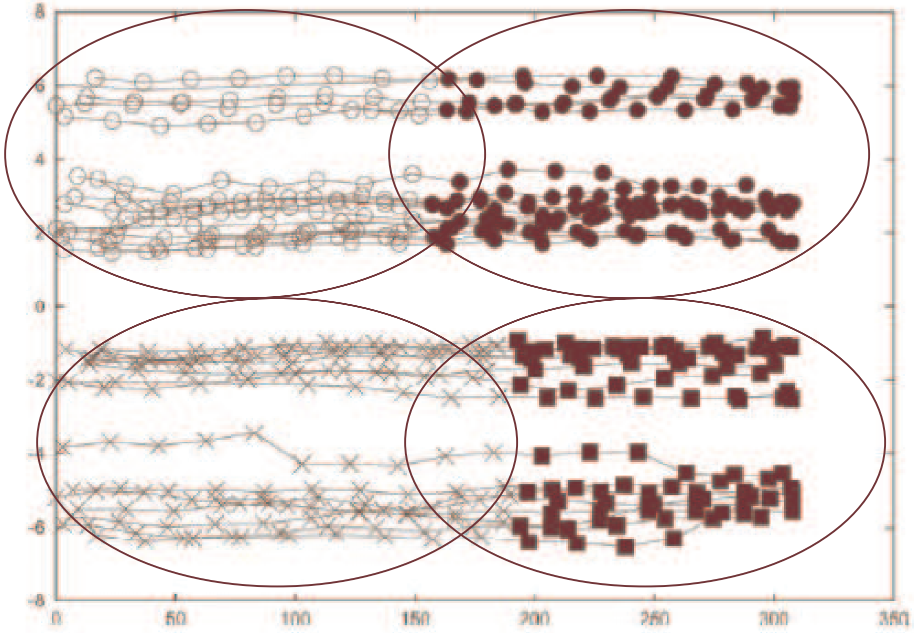
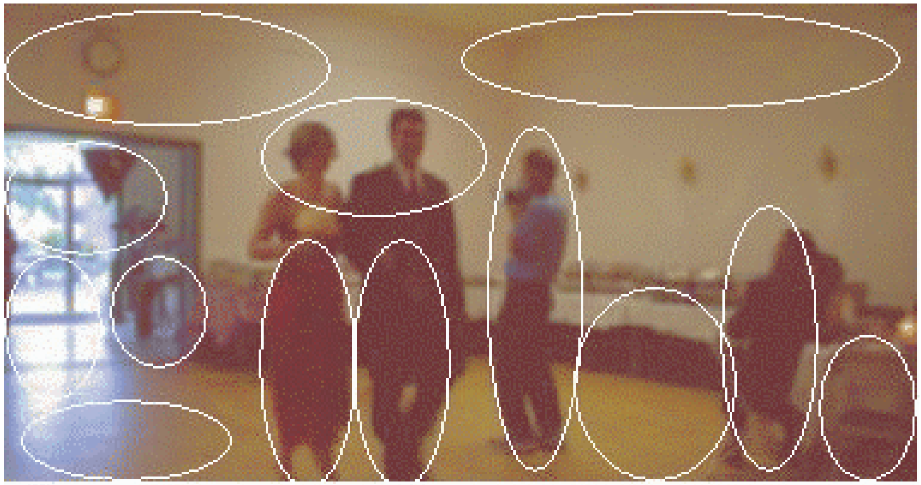
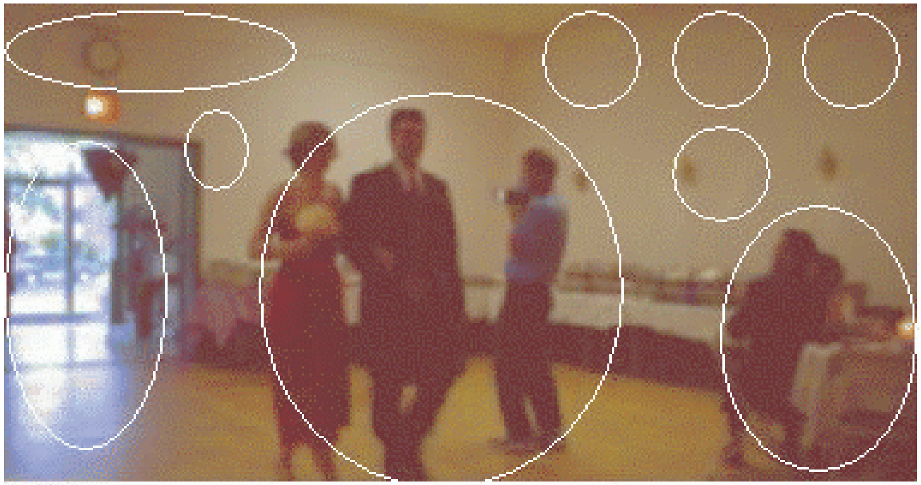


Figure 9. *k*-means output for data set 6, $k = 4$, with nearest clusters marked with different symbols.

Clustering For Object Identification



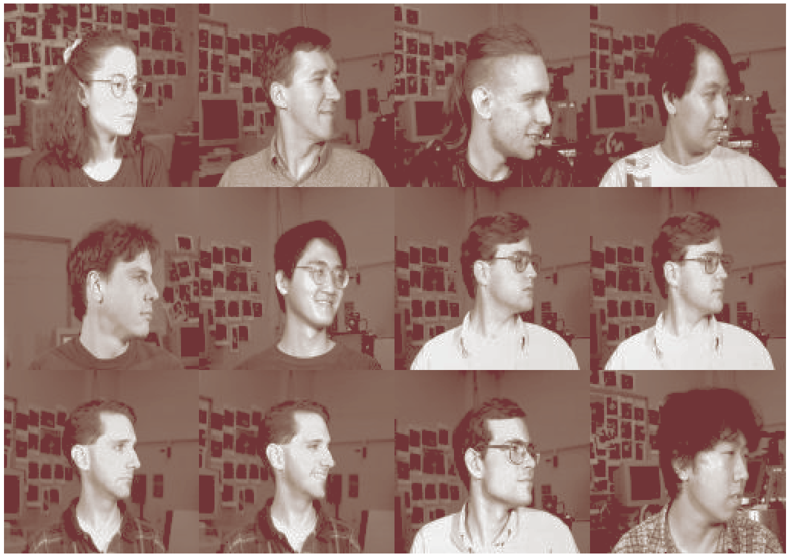
Object identification for Aibo robots

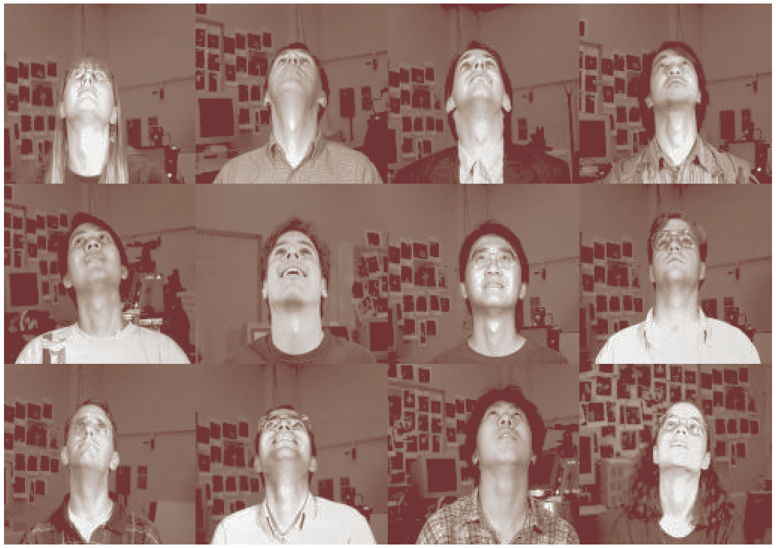


Only significant clusters Shown

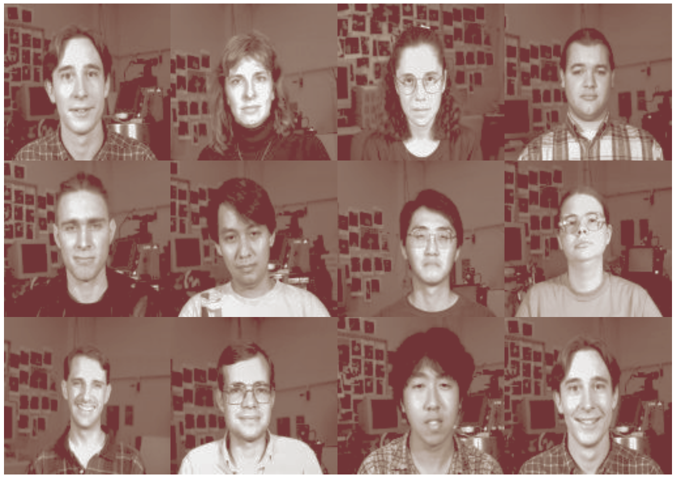
Clustering CMU Faces Database

Example Clusters

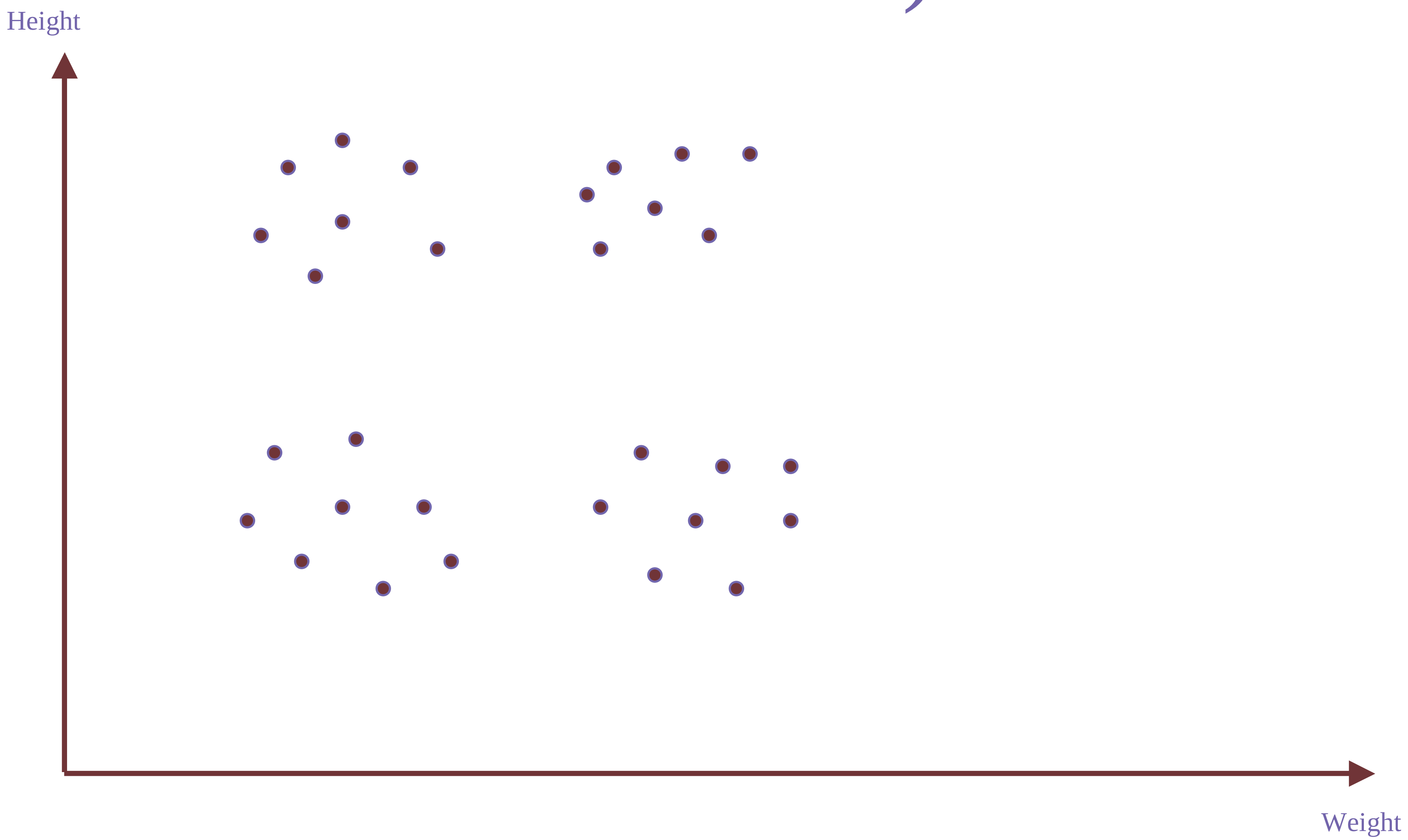




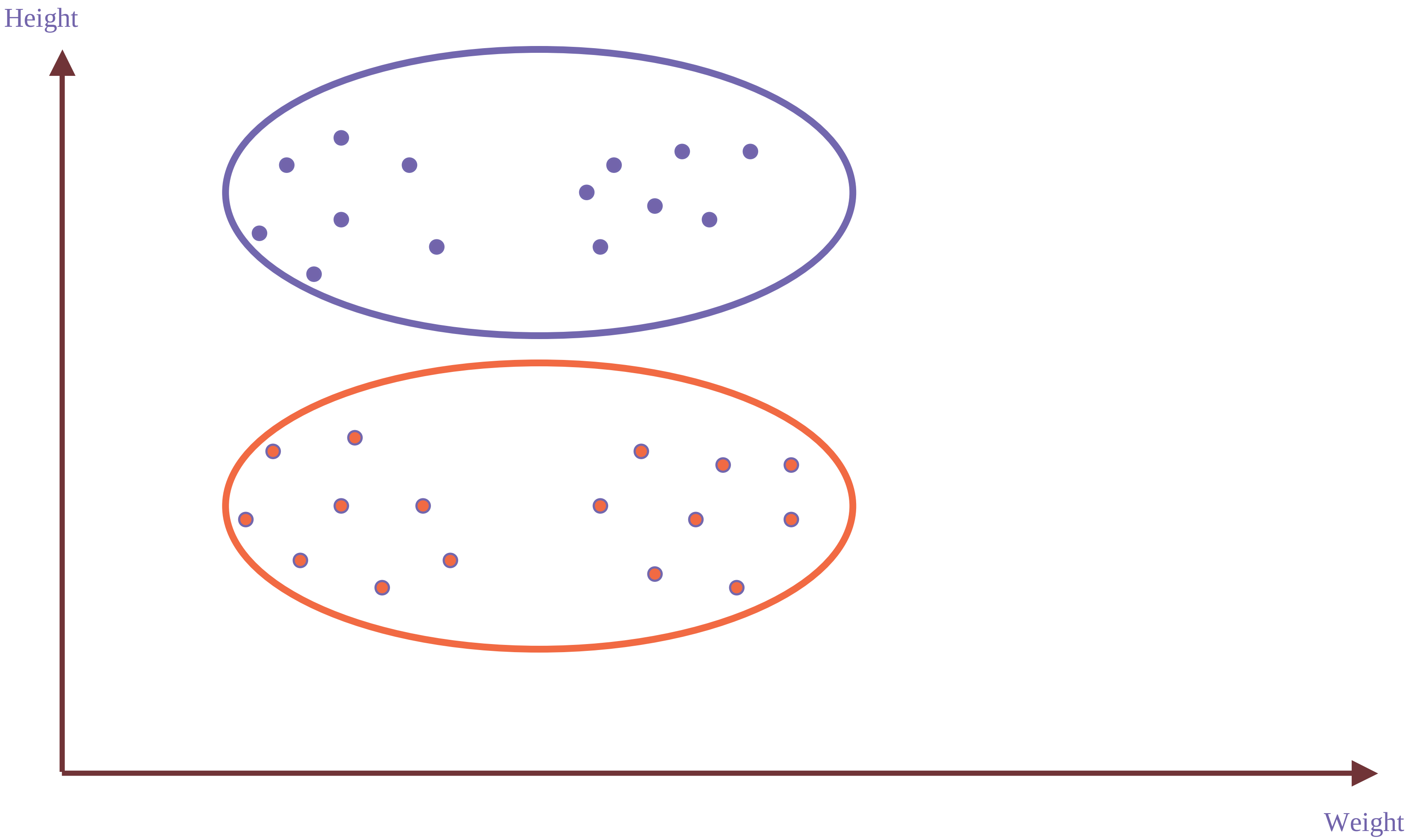
Other Alternatives Beyond Constraints



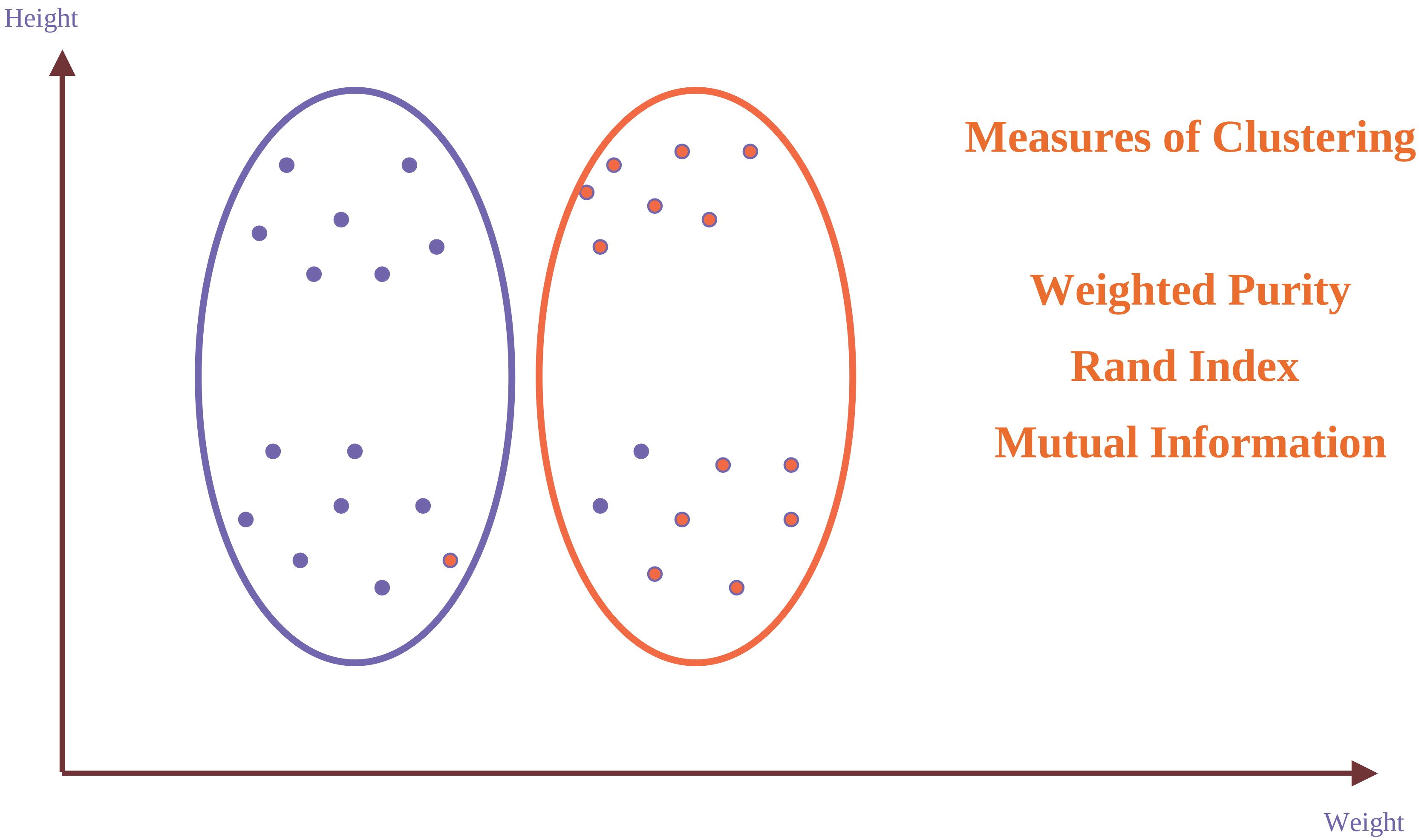
Clustering Example (Number of Clusters=2)



Horizontal Clusters



Vertical Clusters



Measures of Clustering

Weighted Purity

Rand Index

Mutual Information

K-Means Algorithm

1. Randomly assign each instance to a cluster
2. Calculate the centroids for each cluster
3. For each instance
   - Calculate the distance to each cluster center
   - Assign the instance to the closest cluster
4. Goto 2 until distortion is small

K-Means Clustering

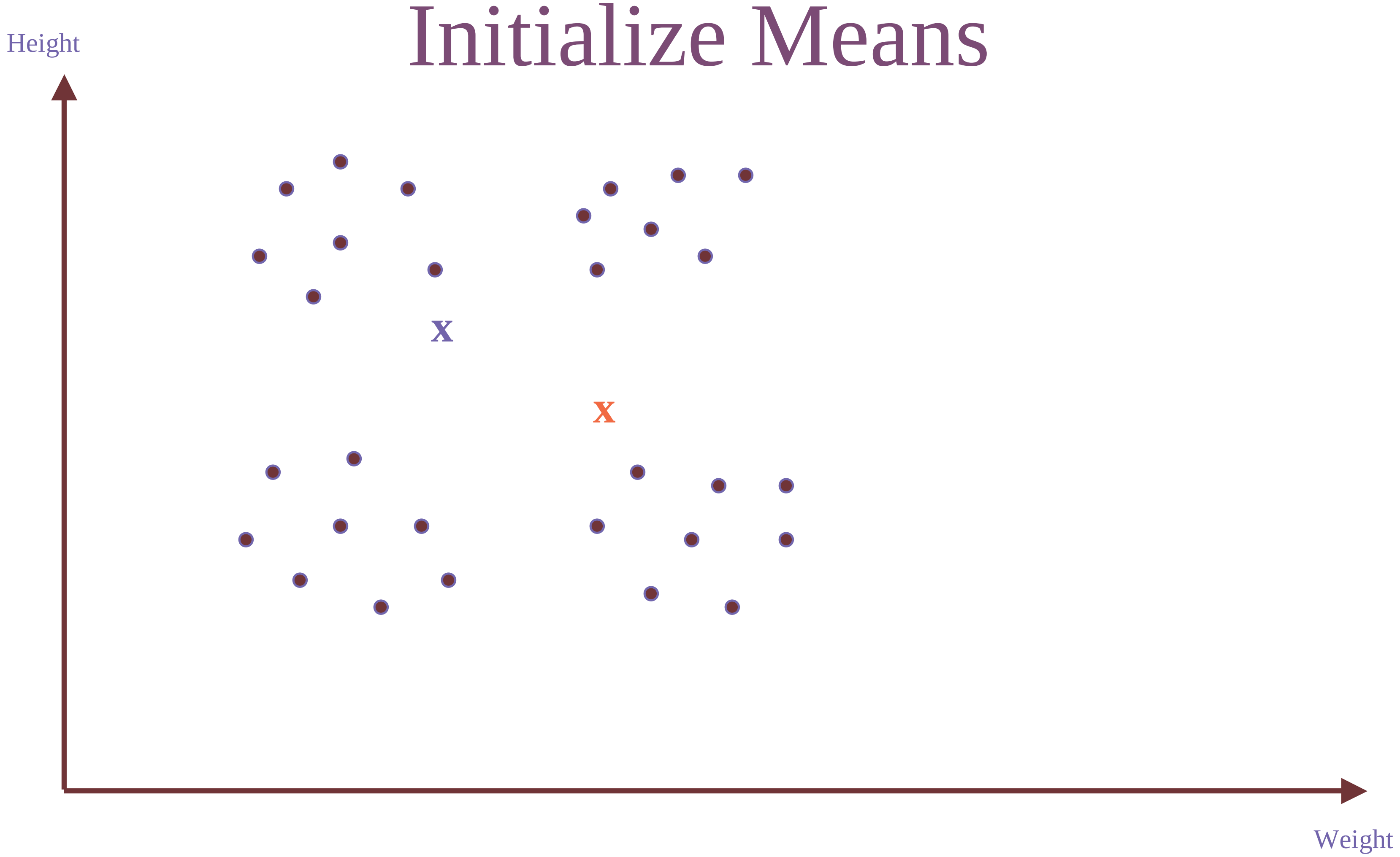
- Standard iterative partitional clustering algorithm
- Finds *k* representative centroids in the dataset
  - Locally minimizes the sum of distance (e.g., squared Euclidean distance) between the data points and their corresponding cluster centroids

$$\sum_{s_i \in S} D(s_i, C_{l_i})$$

A simplified form of this problem is intractable [Garey et al.'82]

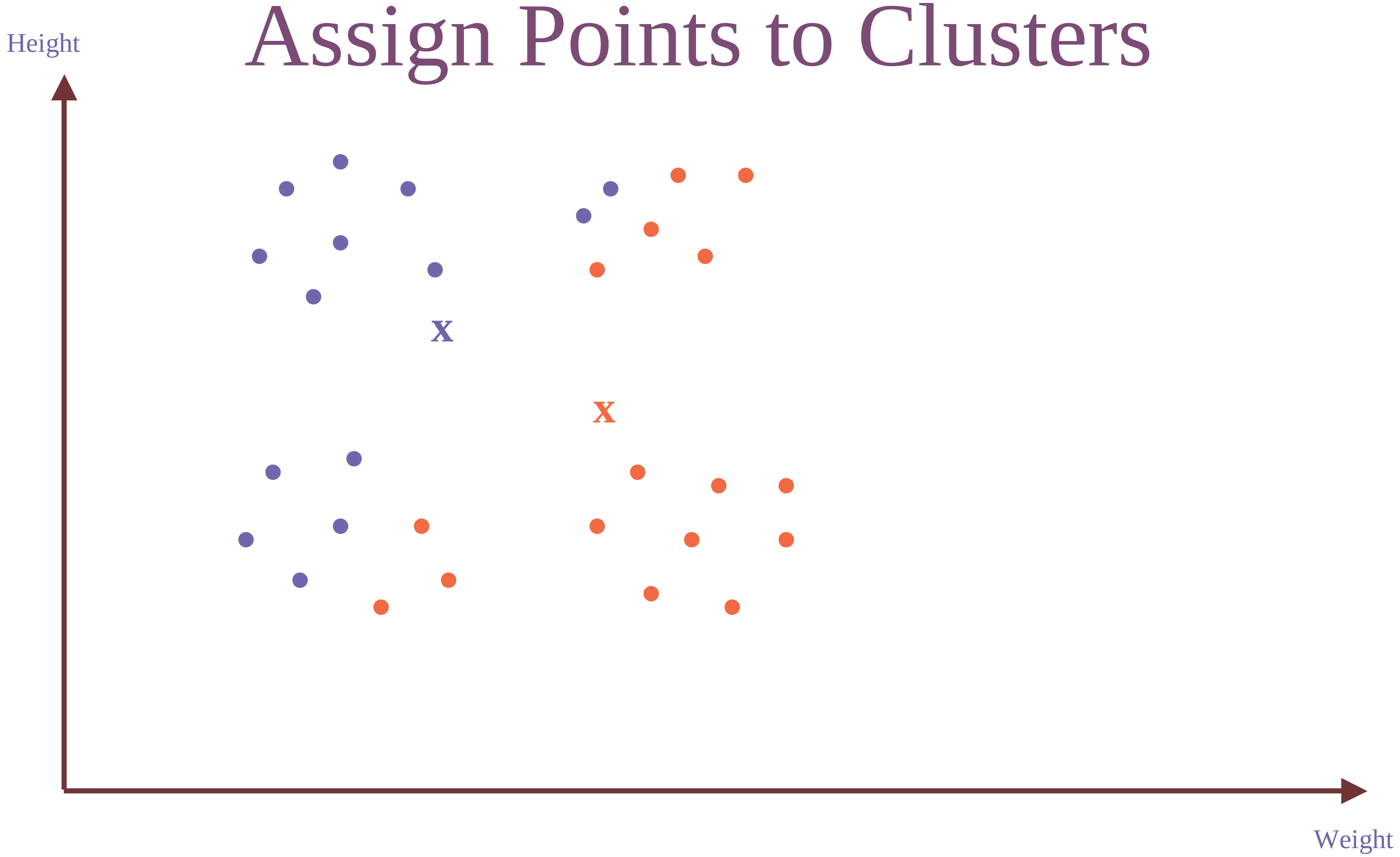
K Means Example (k=2)

Initialize Means



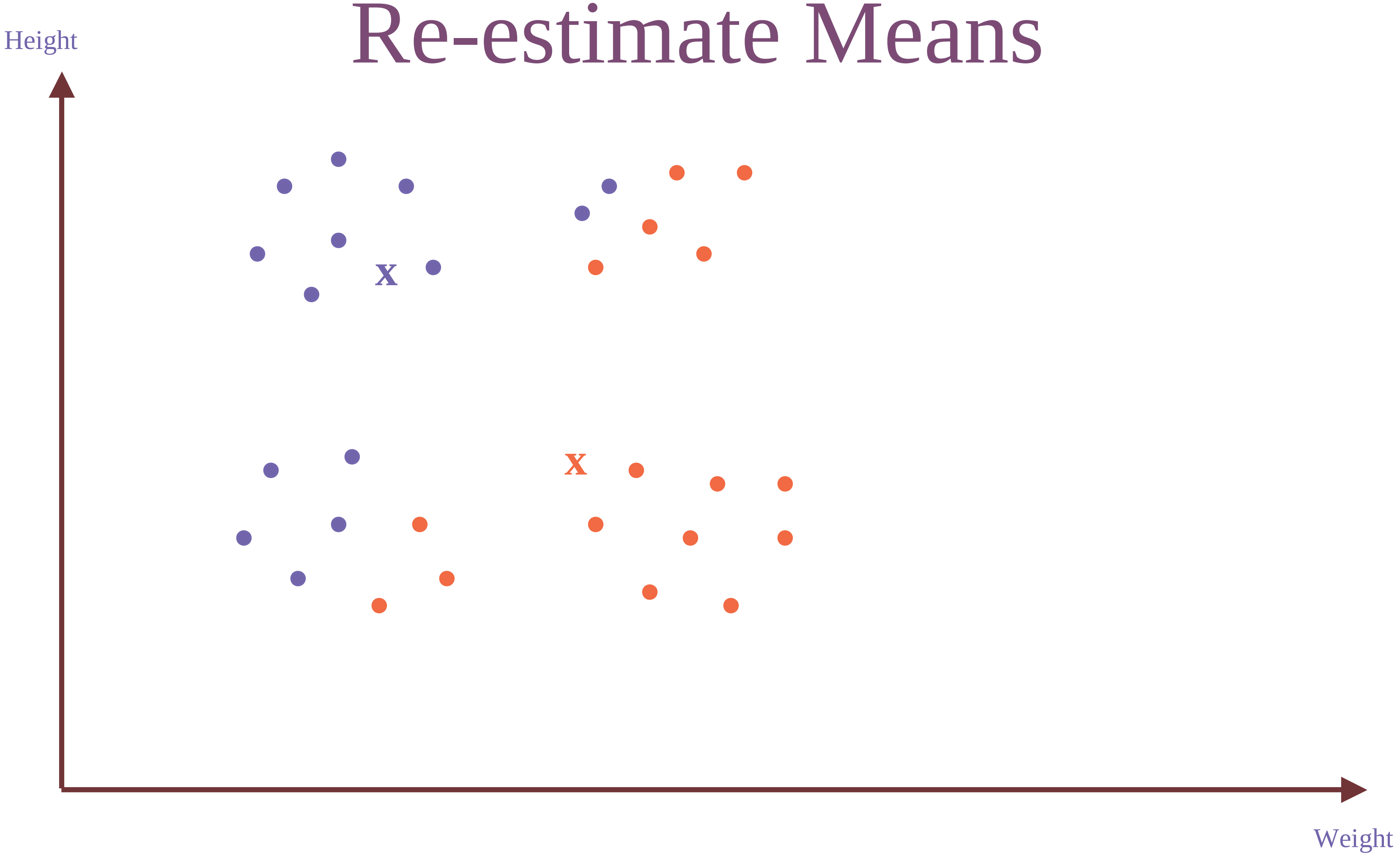
K Means Example

Assign Points to Clusters



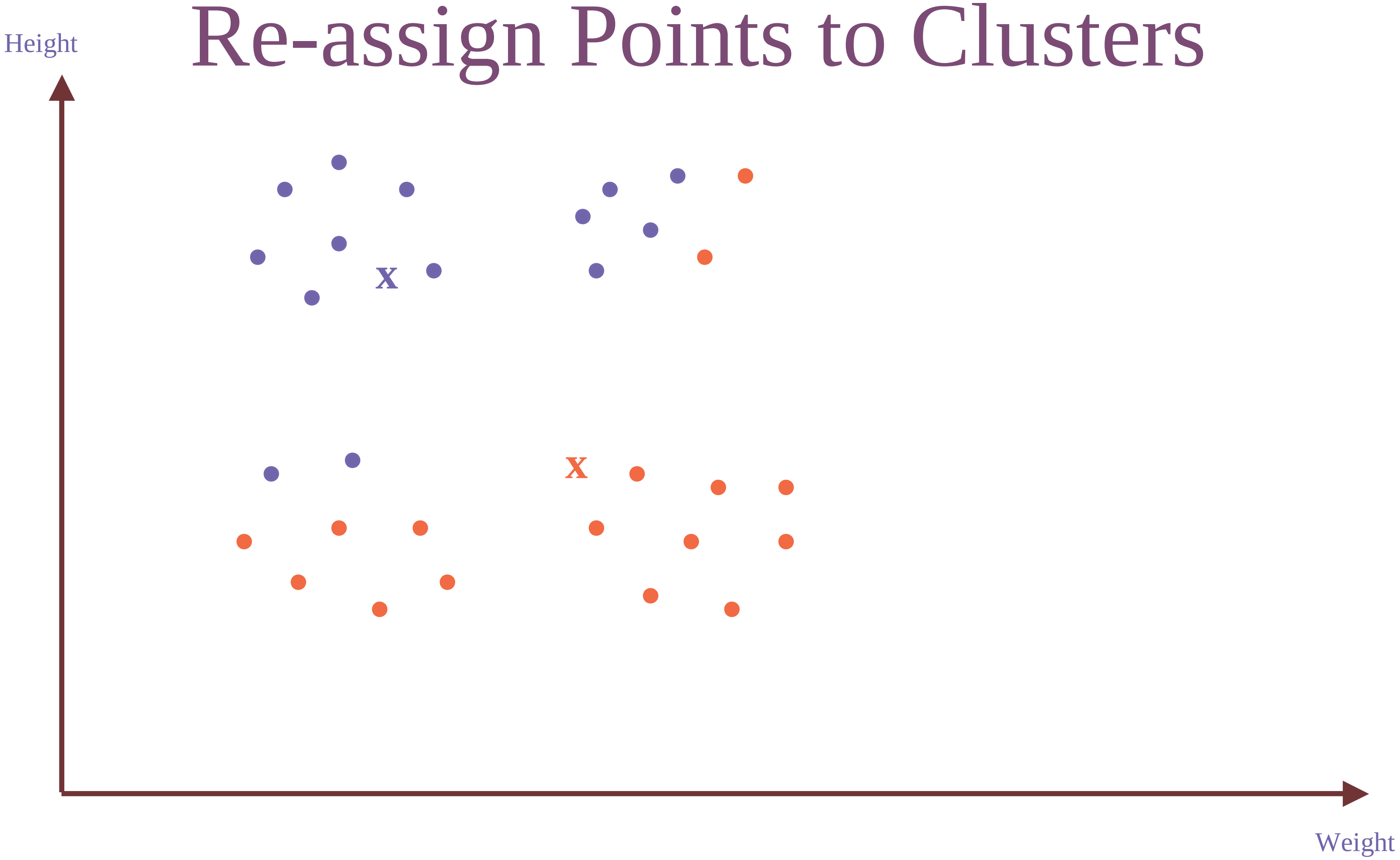
K Means Example

Re-estimate Means



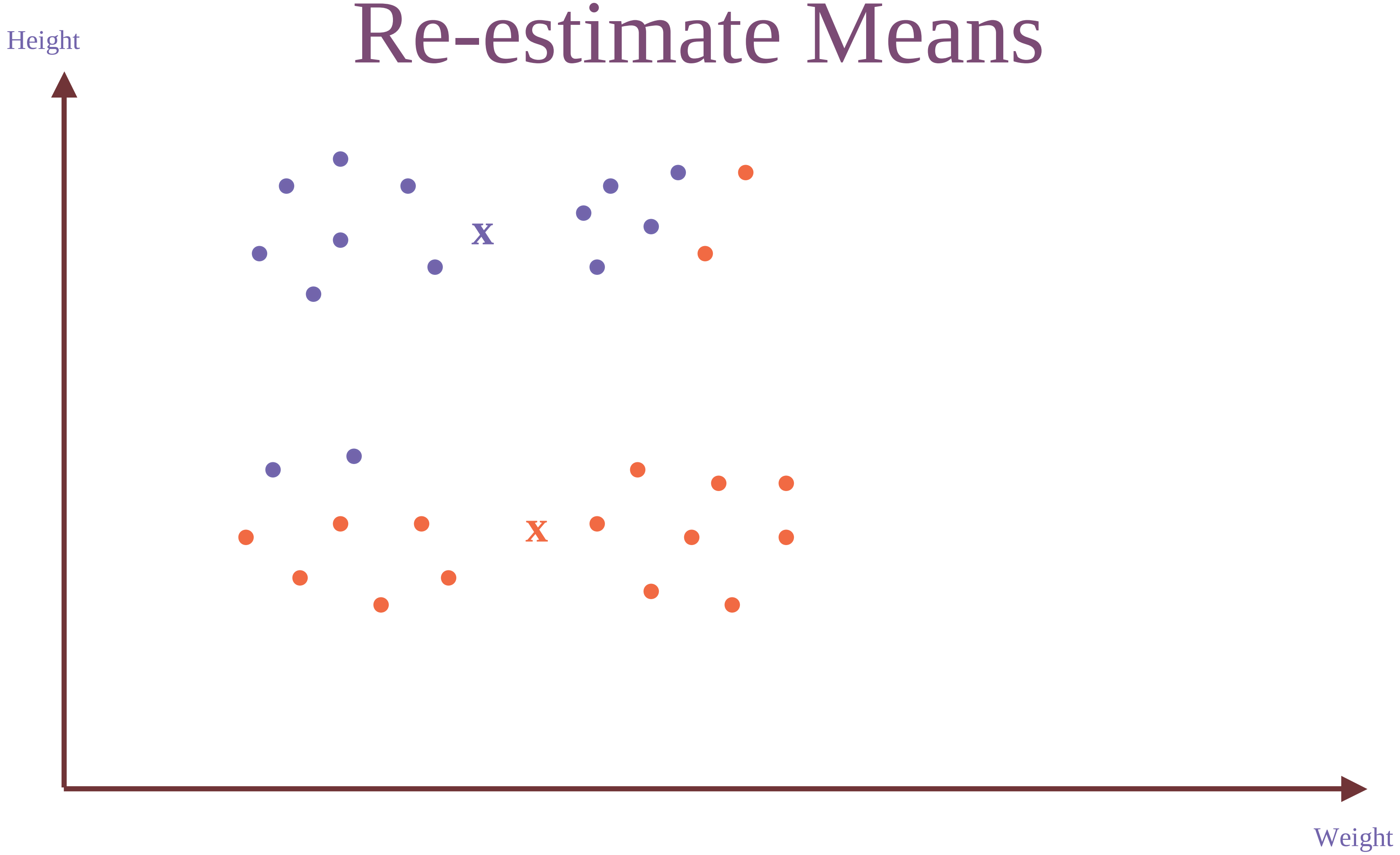
K Means Example

Re-assign Points to Clusters



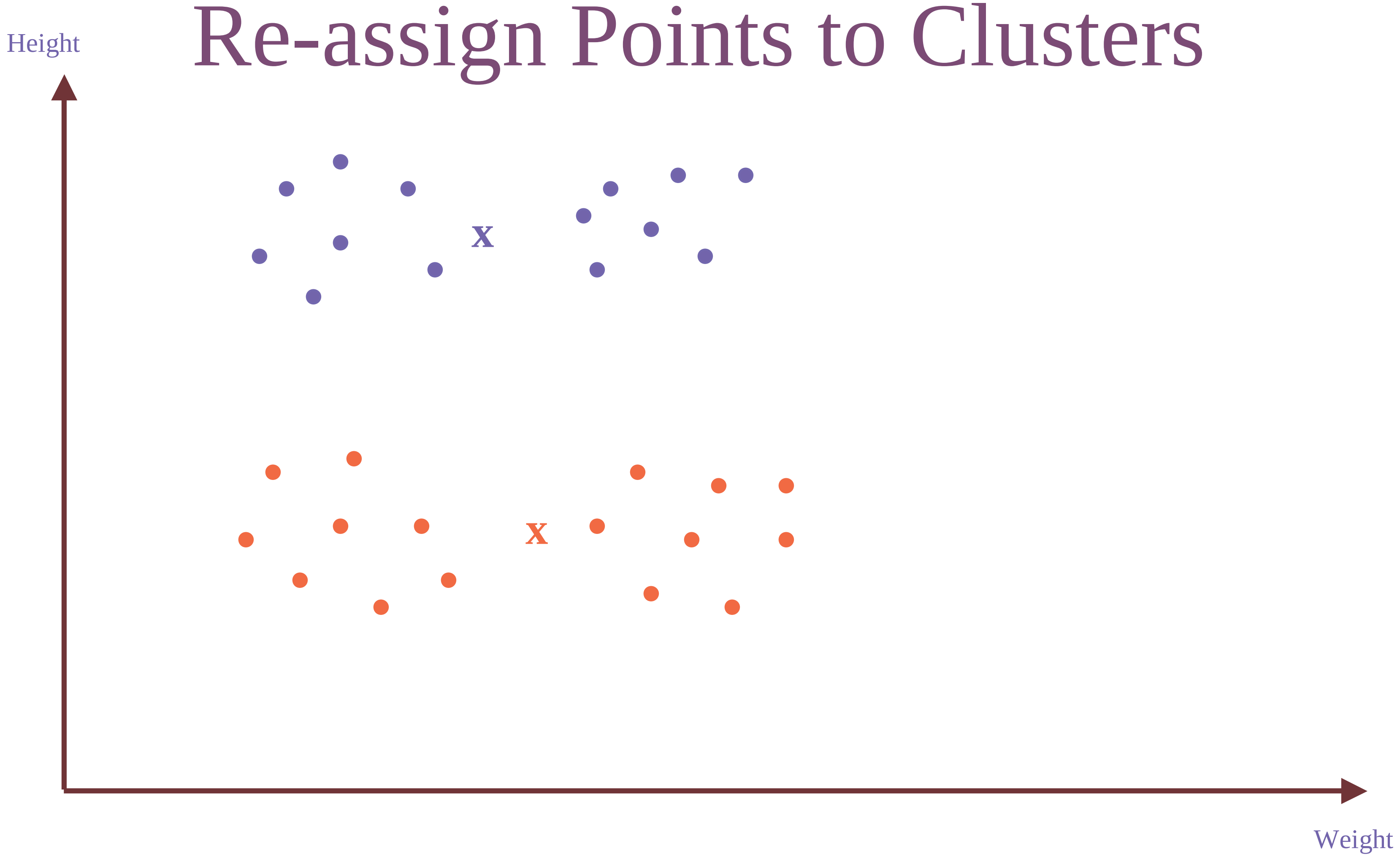
K Means Example

Re-estimate Means



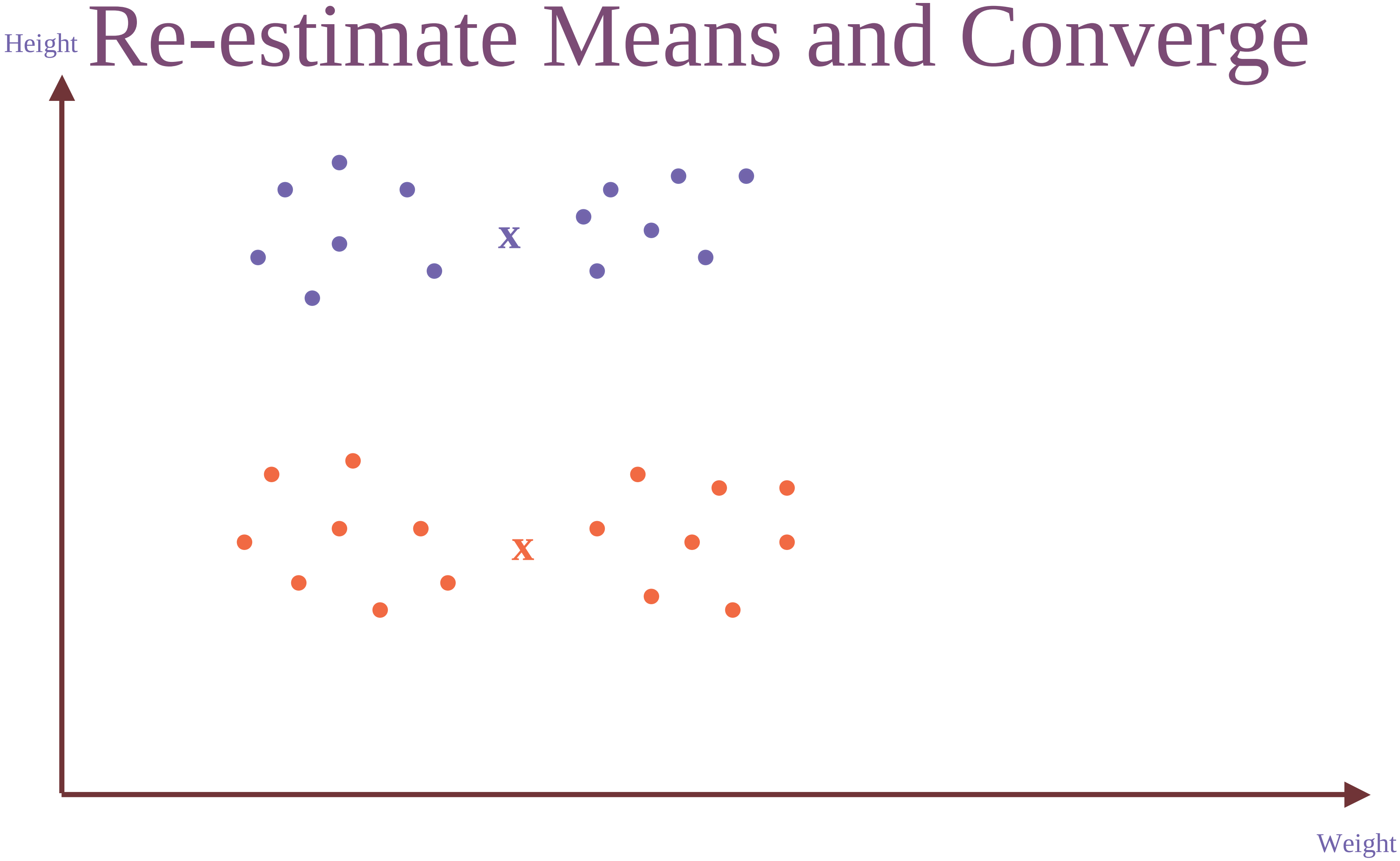
K Means Example

Re-assign Points to Clusters



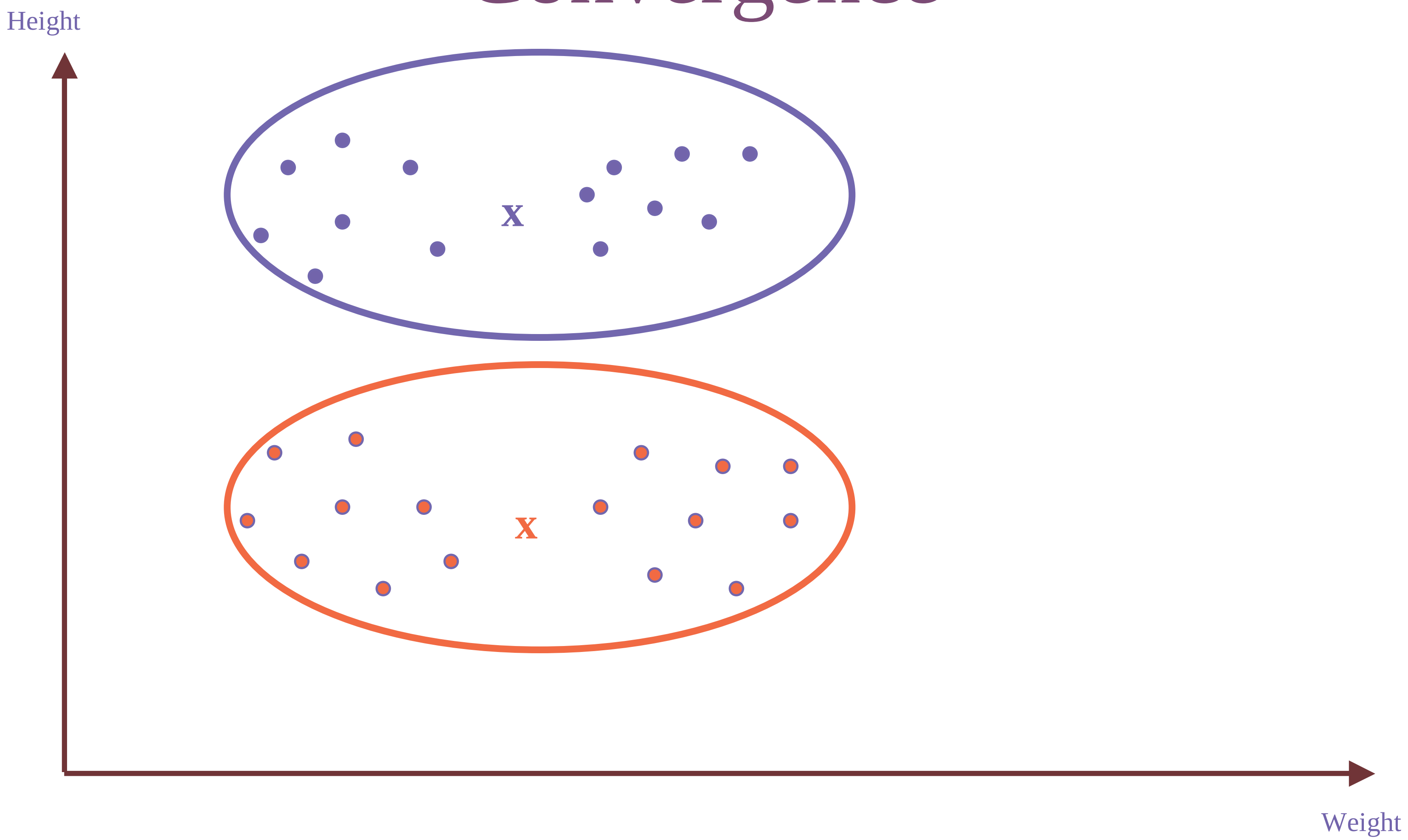
K Means Example

Re-estimate Means and Converge



K Means Example

Convergence



Basic Instance Level Constraints

- Historically, instance level constraints motivated by the availability of labeled data
  - i.e., much unlabeled data and a little labeled data available generally as constraints, e.g., in web page clustering
- This knowledge can be encapsulated using instance level constraints [Wagstaff et al. ’01]
  - Must-Link Constraints
    - A pair of points s_i and s_j $(i \neq j)$ must be assigned to the same cluster.
  - Cannot-Link Constraints
    - A pair of points s_i and s_j $(i \neq j)$ can not be assigned to the same cluster.

Properties of Instance Level Constraints

- Transitivity of Must-link Constraints
  - $ML(a,b)$ and $ML(b,c) \rightarrow ML(a,c)$
  - Let X and Y be sets of points connected by ML constraints
  - $ML(X)$ and $ML(Y)$, $a \in X$, $a \in Y$, $ML(a,b) \rightarrow ML(X \cup Y)$

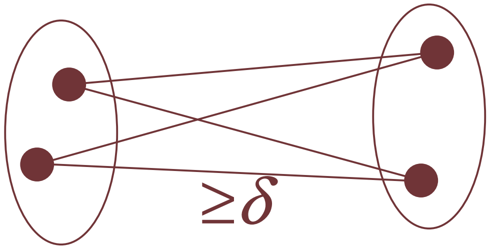
- The Entailment of Cannot link Constraints
  - $ML(a,b)$, $ML(c,d)$ and $CL(a,c) \rightarrow CL(a,d),CL(b,c),CL(b,d)$
  - *Let* $CC_1 \ldots CC_r$ be the groups of must-linked instances (i.e., the connected components)
  - $CL(a \in CC_i\ , b \in CC_j) \rightarrow CL(x,y),\ \forall x \in CC_i,\ \forall y \in CC_j$

Complex Cluster Level Constraints

- δ-Constraint (Minimum Separation)
  - For any two clusters S_i, S_j $\forall$ i,j
  - For any two instances $s_p \in S_i$, $s_q \in S_j$ $\forall$ p,q
  - $D(s_{p,}\, s_q) \geq \delta$
- ε-Constraint
  - For any cluster S_i $|S_i| > 1$
  - $\forall p,\ s_p \in S_i, \exists s_q \in S_i : \varepsilon \geq D(s_{p,}\, s_q), s_p <> s_q$

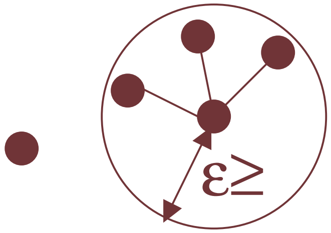
Converting Cluster Level to Instance Level Constraints

- Delta constraints?



For every point *x*, must-link all points *y* such that D(x,y) < *δ*, i.e. conjunction of ML constraints

- Epsilon constraints?
  - For every point *x*, must link to at least one point *y* such that D(x,y) <= *ε*, i.e. disjunction of ML constraints



- Will generate many instance level constraints

Other Constraint Types We Won’t Have Time To Cover

- Balanced Clusters
  - Scalable model-based balanced clustering [Zhong et al. ’03]
  - Frequency sensitive competitive learning [Galanopoulos et al. ’96, Banerjee et al. ’03]
  - K-Means clustering with cluster size constraints [Bradley et al. ’00]
- Clustering only with constraints
  - Correlation Clustering / Clustering with Qualitative Information [Bansal et al.’02, Charikar et al. ’03, Blum et al. ’04, Demaine et al.]
  - No distance function, use only constraints to cluster data
  - Maximize the agreements / minimize disagreements between cluster partitioning and constraints

Other Constraint Types We Won’t Have Time To Cover

- Negative background information
  - Find another clustering that is quite different from a given set of clusterings [Gondek et al. ’04]
- Labels given on data subset
  - Genetic algorithm to incorporate labeled supervision [Demiriz et al.’00]
  - Modify cluster assignment step to satisfy given labels [Basu et al.’02]
  - Cluster using conditional distributions of labels in an auxilliary space [Sinkkonen et al. ’04]
  - Fit Bayesian model with Dirichlet Process prior [Daume et al.’05]
    - learns appropriate number of clusters using non-parametric technique
- Attribute-level / model-level constraints [Law et al.’05]

Outline

- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Big Picture

- Clustering with constraints:

  Partition unlabeled data into groups called clusters

  + use constraints to aid and bias clustering

- Goal:

  Examples in same cluster similar, separate clusters different + constraints are maximally respected

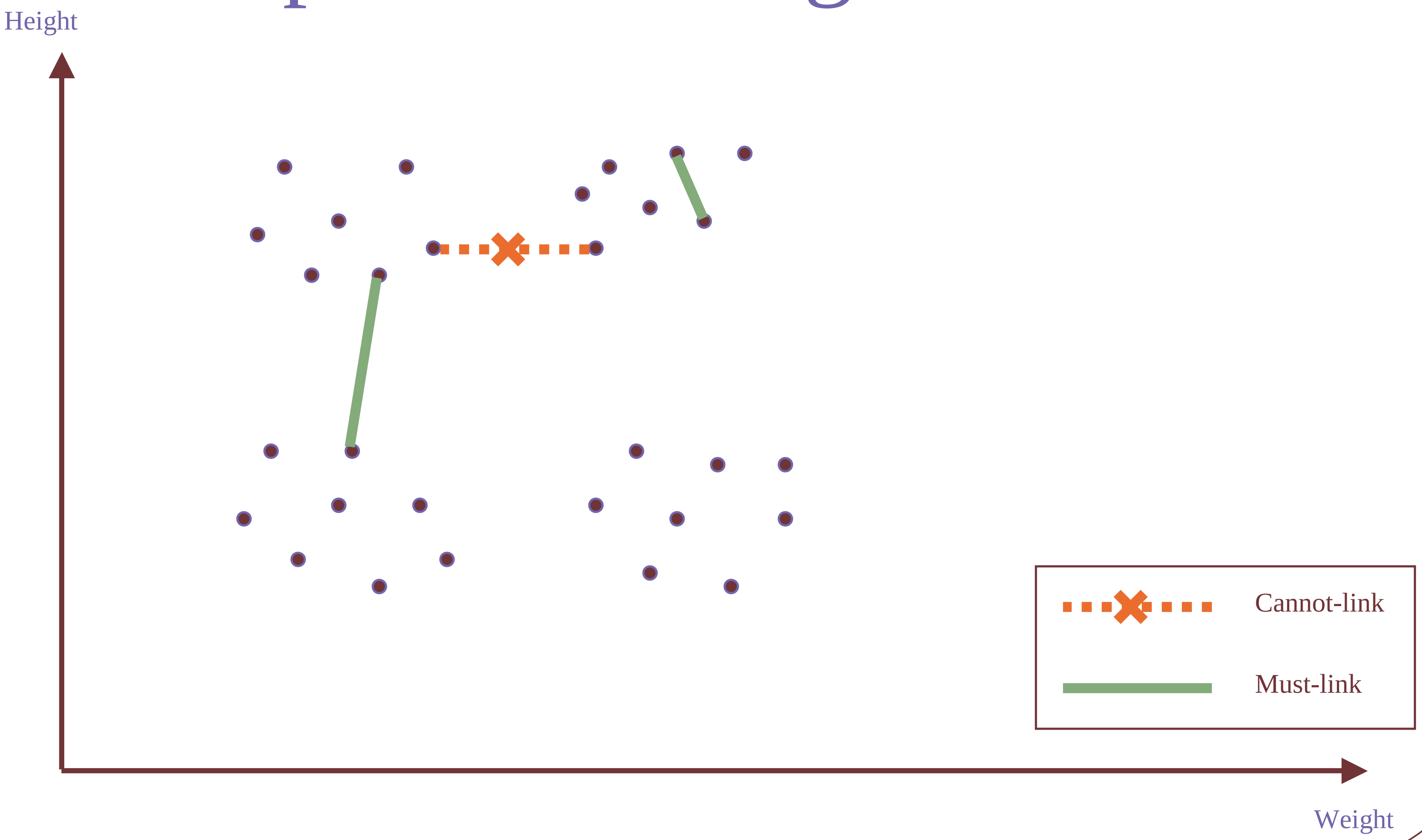
Enforcing Constraints

- Clustering objective modified to enforce constraints
  - Strict enforcement: find “best” feasible clustering respecting all constraints
  - Partial enforcement: find “best” clustering maximally respecting constraints

- Uses standard distance functions for clustering

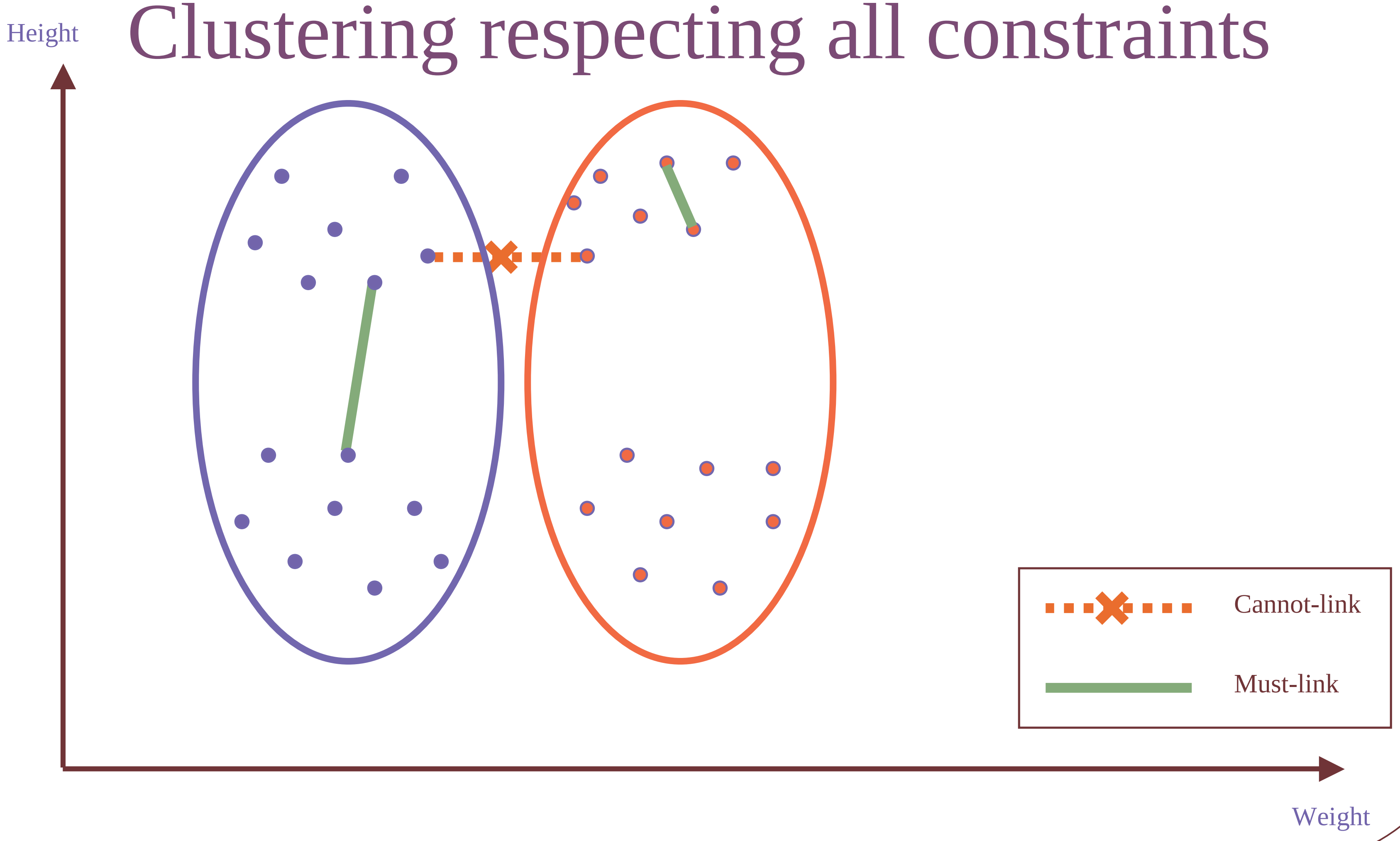
[Demiriz et al.’99, Wagstaff et al.’01, Segal et al.’03, Davidson et al.’05, Lange et al.’05]

Example: Enforcing Constraints



Example: Enforcing Constraints

Clustering respecting all constraints

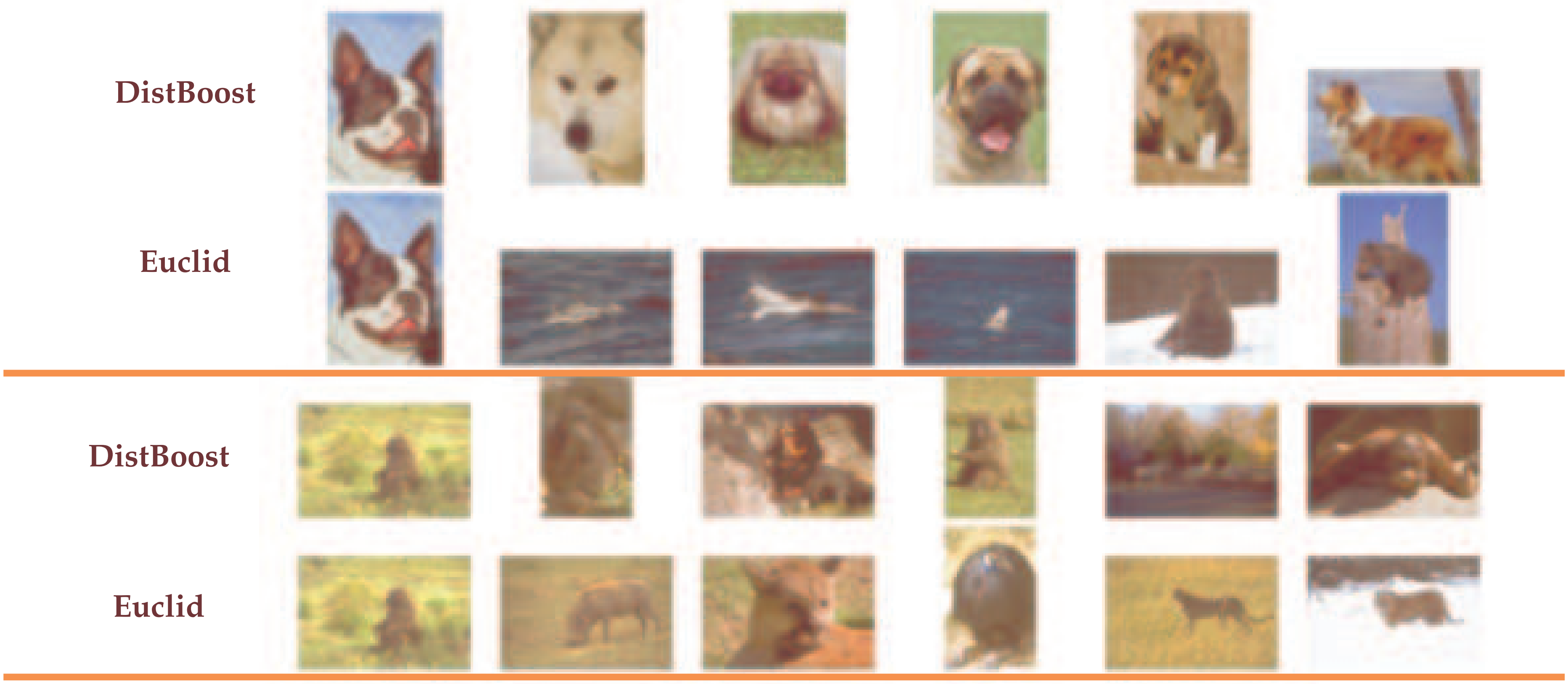


Learning Distance Function

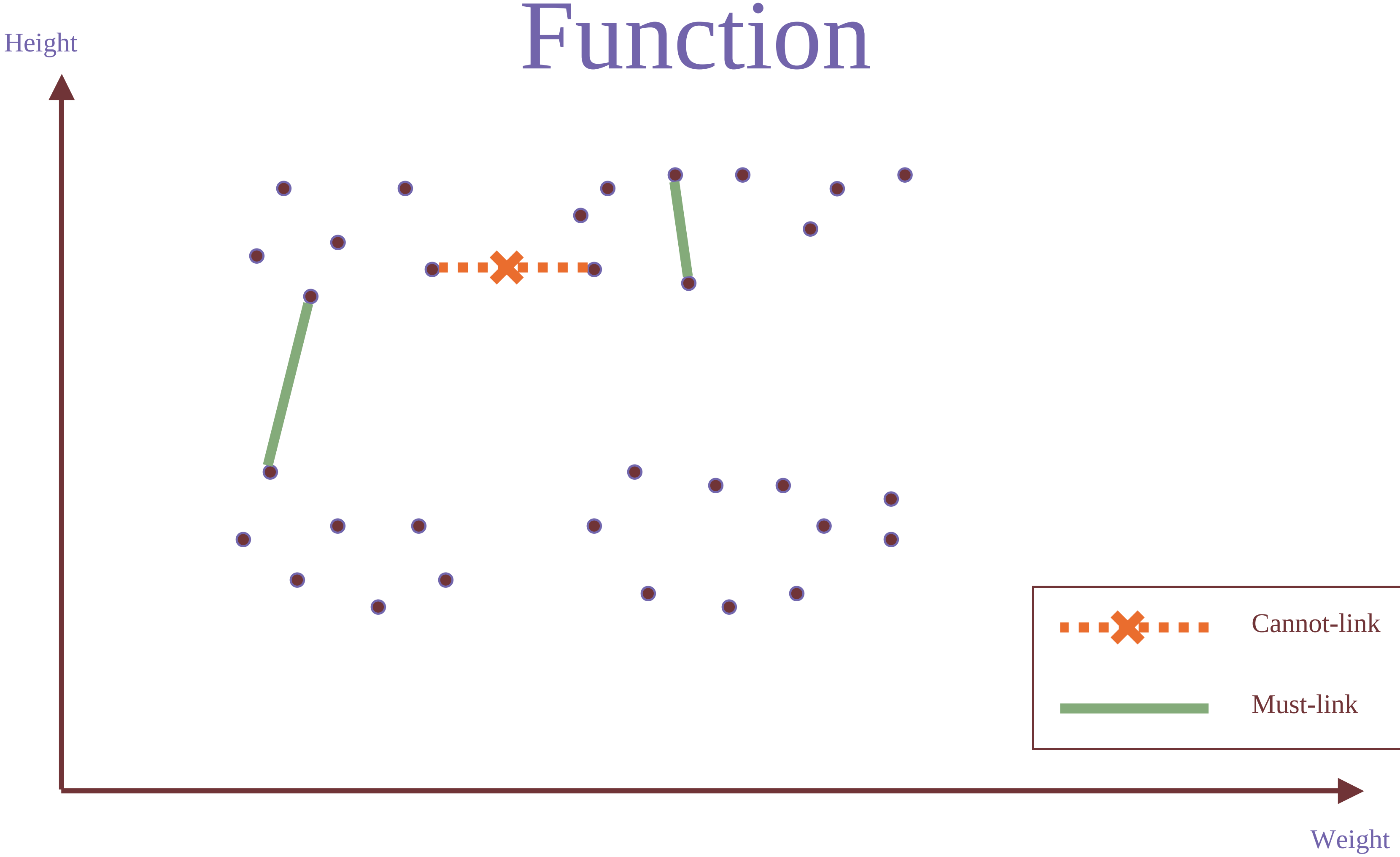
- Constraints used to learn clustering distance function
  - *ML(a,b)* → *a* and *b* and surrounding points should be "close"
  - *CL(a,b)* → *a* and *b* and surrounding points should be "far apart"
- Standard clustering algorithm applied with learned distance function

[Klein et al.'02, Cohn et al.'03, Xing et al.'03, Bar Hillel et al.'03, Bilenko et al.'03, Kamvar et al.'03, Hertz et al.'04, De Bie et al.'04]

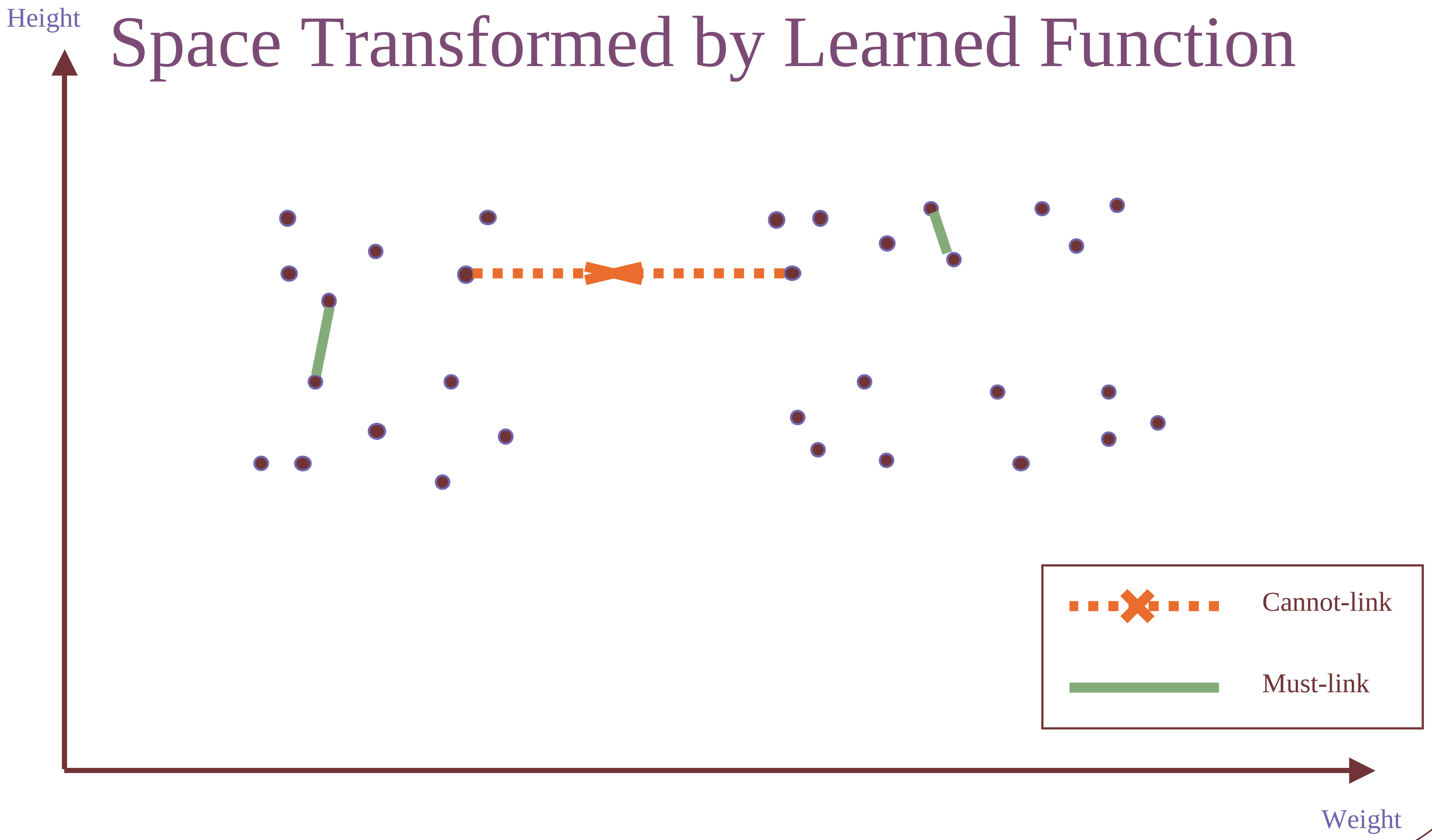
Why Learn Distance Functions?



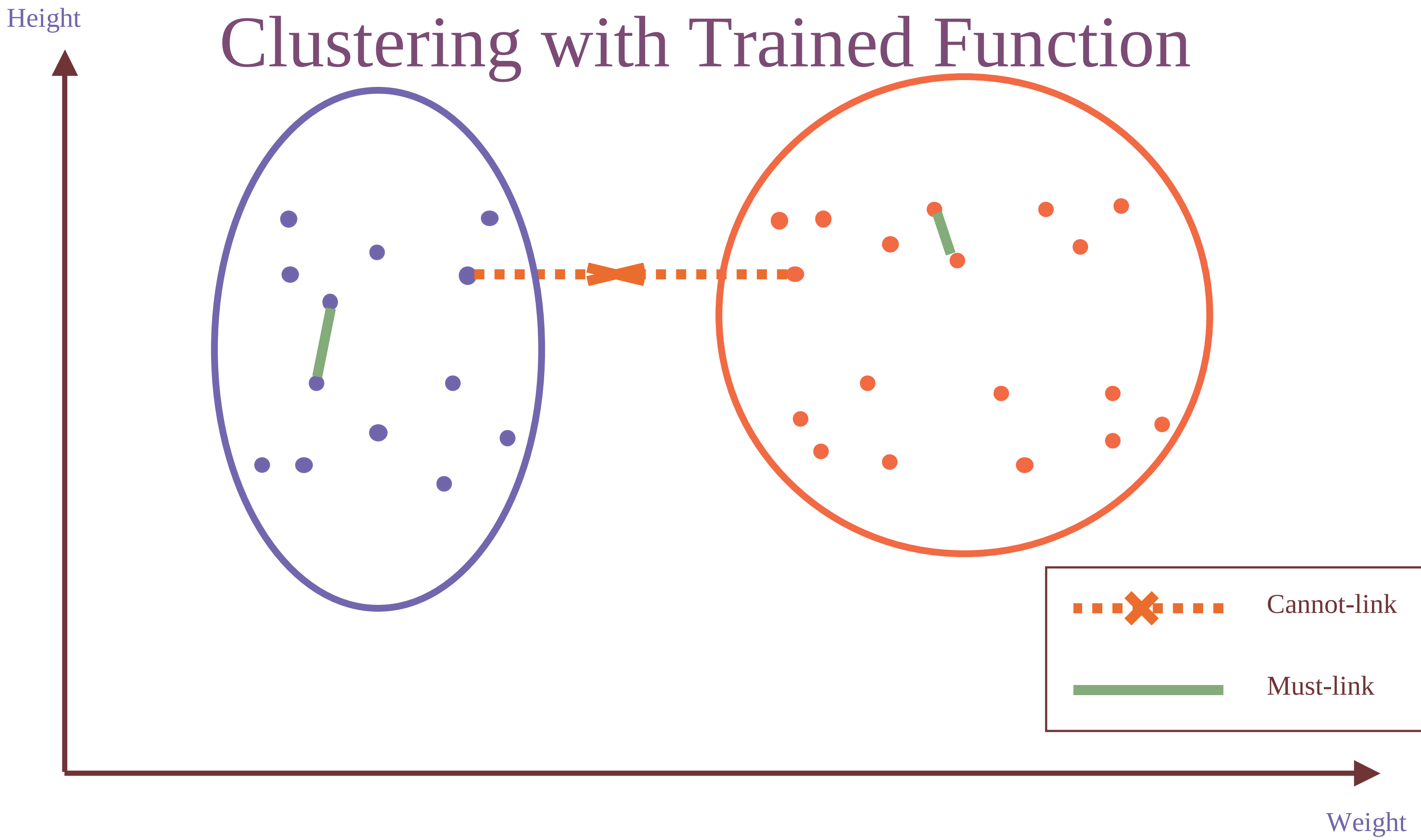
Example: Learning Distance Function



Example: Learning Distance Function



Example: Learning Distance Function



Enforce Constraints + Learn Distance

- Integrated framework [Basu et al.'04]
  - Respect constraints during cluster assignment
  - Modify distance function during parameter re-estimation

- Advantage of integration
  - Distance function can change the space to decrease constraint violations made by cluster assignment
  - Uses both constraints and unlabeled data for learning distance function

Outline

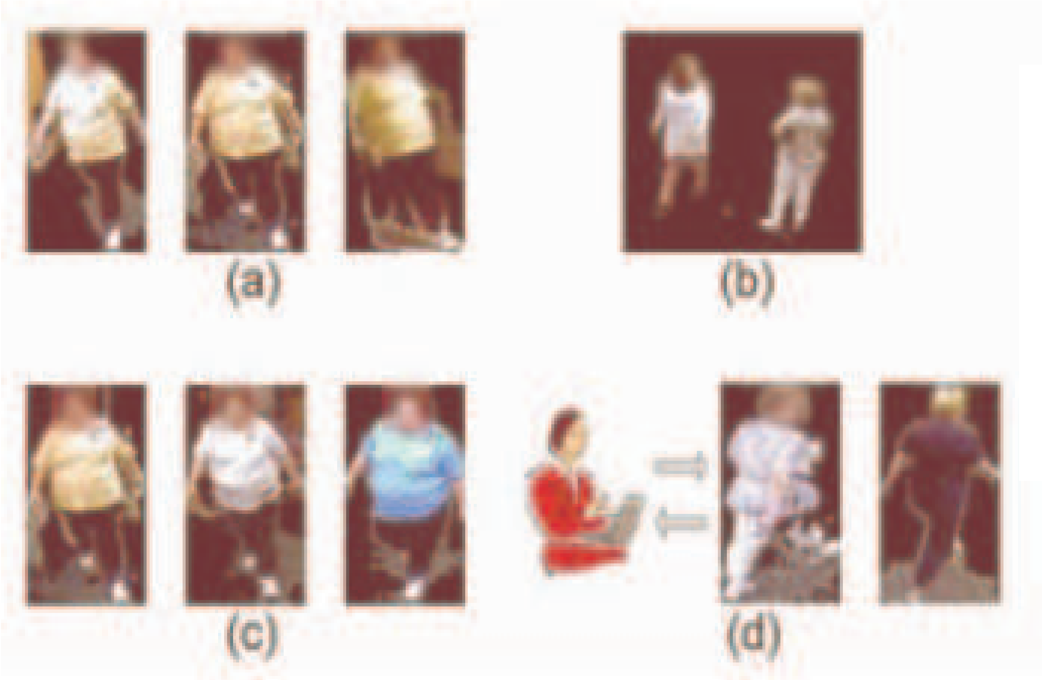
- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
    - Enforcing constraints [Ian]
    - Hierarchical [Ian]
    - Learning distances [Sugato]
    - Initializing and pre-processing [Sugato]
    - Graph-based [Sugato]

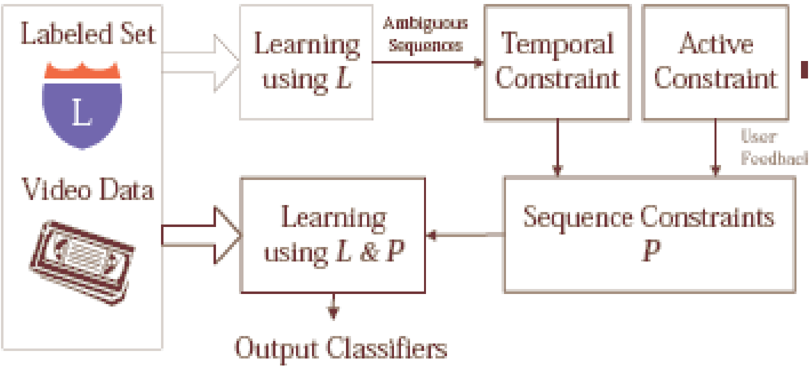
Generating Constraints From Labels

- Most used (in papers) approach to generate constraints.
- Typically set *k* to equal the number of extrinsic classes
- Clustering labeled (D_l) and unlabeled data (D_u)
- Generate constraints from D_l (but how much?, what happens if I generate too many constraints?)

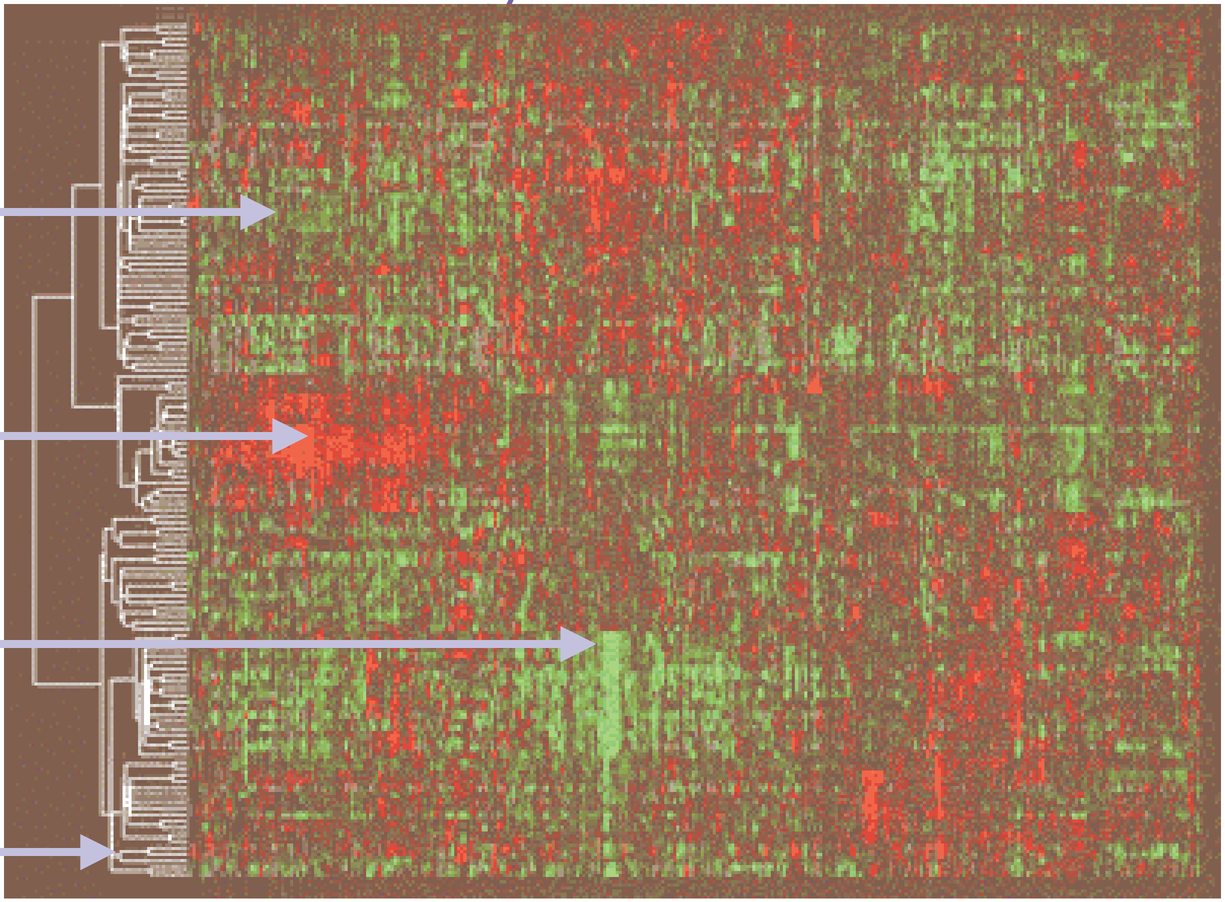
Generating Constraints from Video

- Generating constraints from spatio-temporal aspects of video sequences [Yan et al.’04]





Gene Clustering Using Micro-array Data



Gene expressions

Red => low expression w.r.t baseline

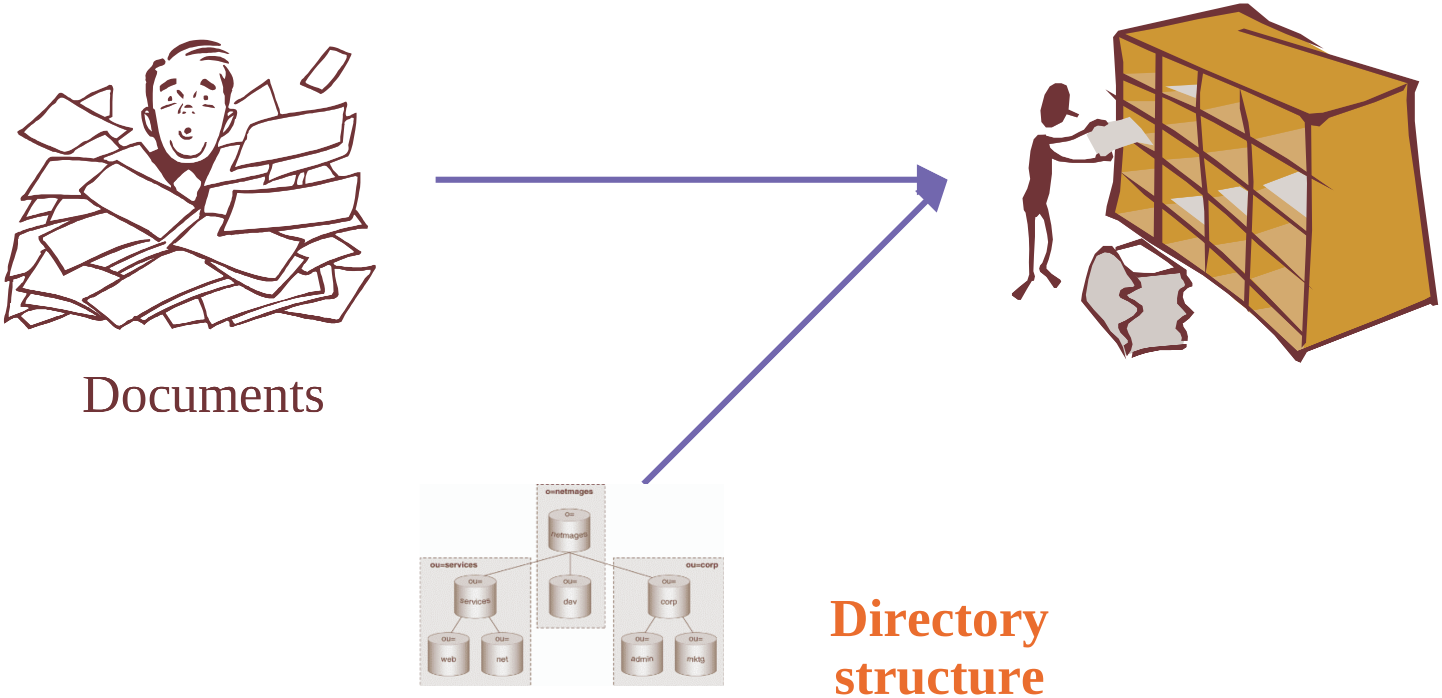
Green => high expression w.r.t baseline

Gene clusters

- **Constraints from gene interaction information in DIP**

Content Management: Document Clustering

Clustering



Documents

Directory structure constraints

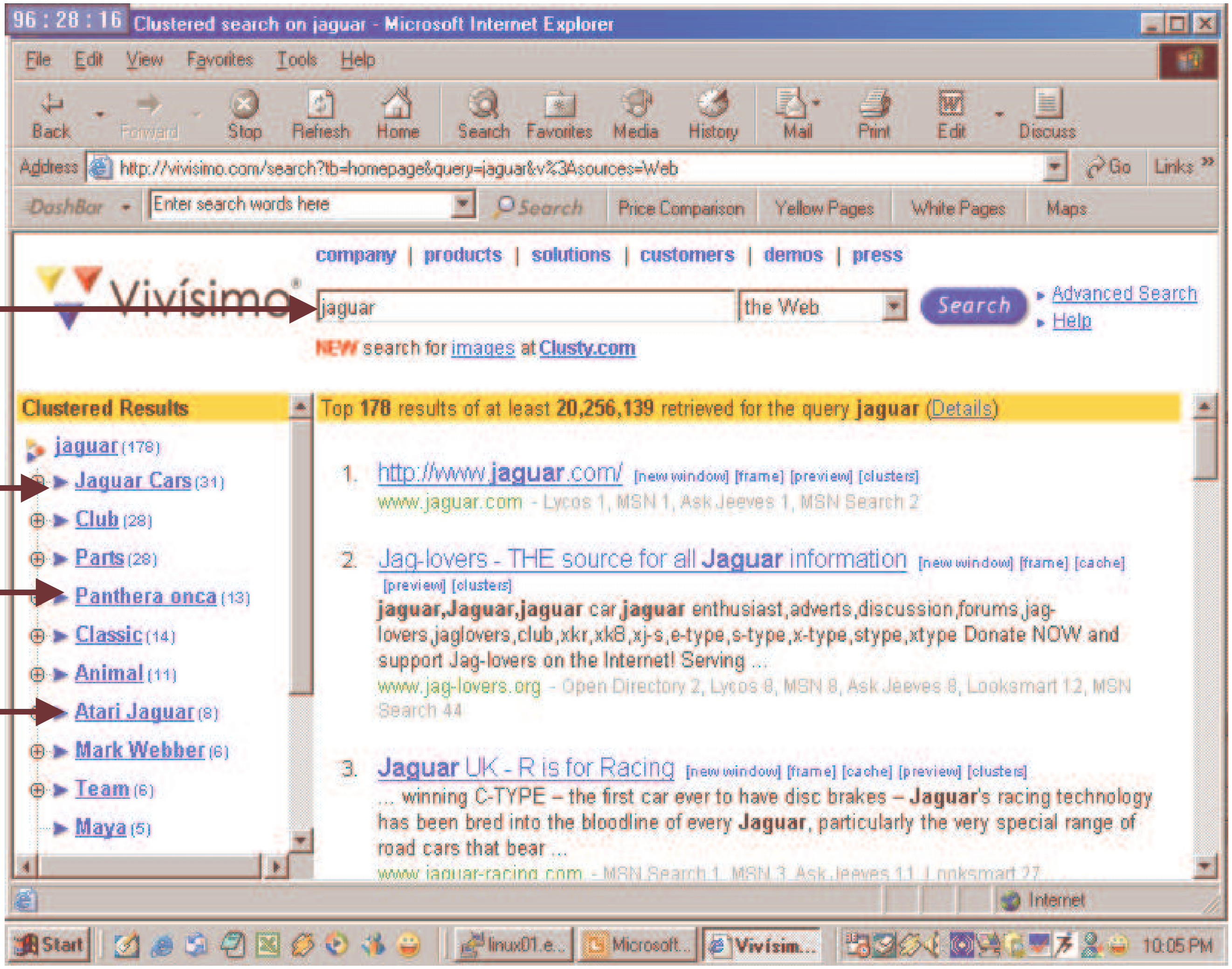
Personalizing Web Search Result Clustering

Query: jaguar

Jaguar cars

Jaguar animal

Macintosh OS X (Jaguar)



96 : 28 : 16 Clustered search on jaguar - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss
Address http://vivisimo.com/search?tb=homepage&query=jaguar&v%3Asources=Web
Go Links
DashBar Enter search words here Search Price Comparison Yellow Pages White Pages Maps
company | products | solutions | customers | demos | press
Vivísimo
jaguar
the Web
Search
Advanced Search
Help
NEW search for images at Clusty.com
Clustered Results
jaguar (178)
Jaguar Cars (31)
Club (28)
Parts (28)
Panthera onca (13)
Classic (14)
Animal (11)
Atari Jaguar (8)
Mark Webber (6)
Team (6)
Maya (5)
Internet
Start linux01.e... Microsoft... Vivísim... 10:05 PM

Top 178 results of at least 20,256,139 retrieved for the query jaguar (Details)

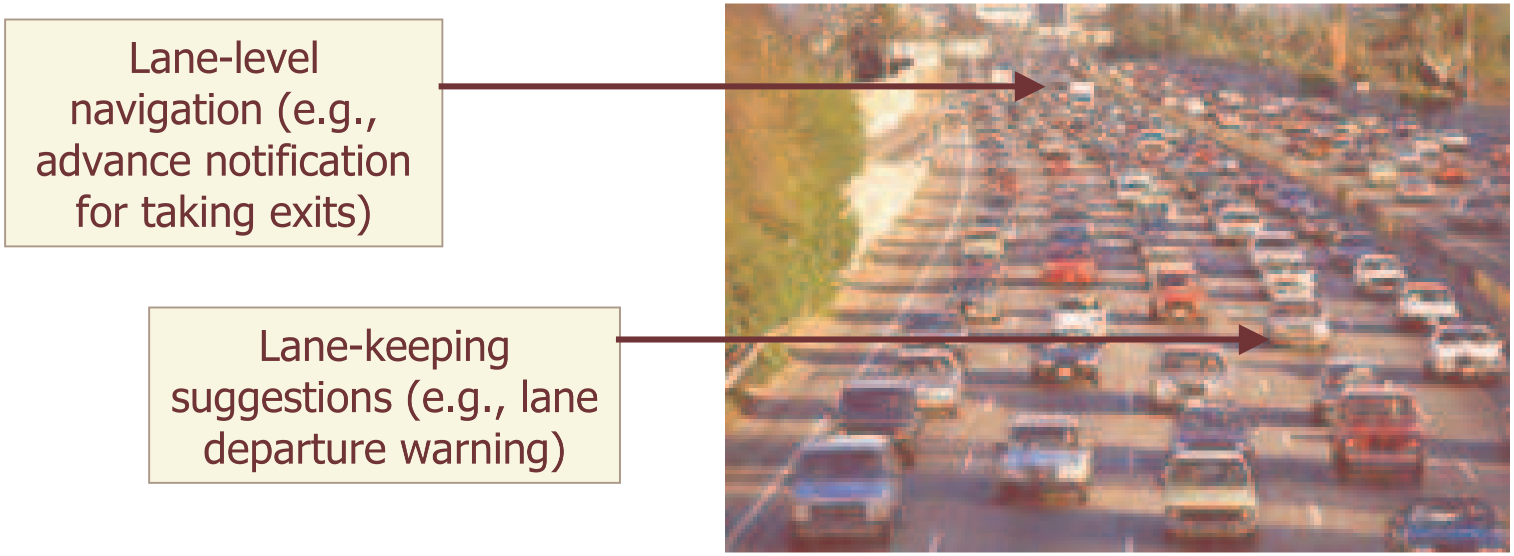
1. http://www.jaguar.com/ [new window] [frame] [preview] [clusters]
www.jaguar.com - Lycos 1, MSN 1, Ask Jeeves 1, MSN Search 2

2. Jag-lovers - THE source for all Jaguar information [new window] [frame] [cache] [preview] [clusters]
jaguar,Jaguar,jaguar car,jaguar enthusiast,adverts,discussion,forums,jag-lovers,jaglovers,club,xkr,xk8,xj-s,e-type,s-type,x-type,stype,xtype Donate NOW and support Jag-lovers on the Internet! Serving ...
www.jag-lovers.org - Open Directory 2, Lycos 8, MSN 8, Ask Jeeves 8, Looksmart 12, MSN Search 44

3. Jaguar UK - R is for Racing [new window] [frame] [cache] [preview] [clusters]
... winning C-TYPE – the first car ever to have disc brakes – Jaguar's racing technology has been bred into the bloodline of every Jaguar, particularly the very special range of road cars that bear ...
www.jaguar-racing.com - MSN Search 1, MSN 3, Ask Jeeves 11, Looksmart 27

- **Constraints mined from co-occurrence information in query web-logs**

Automatic Lane Finding from GPS traces [Wagstaff et al. ’01]



Lane-level navigation (e.g., advance notification for taking exits)

Lane-keeping suggestions (e.g., lane departure warning)

- **Constraints inferred from trace-contiguity (ML) & max-separation (CL)**

Mining GPS Traces (Schroedl et’ al)

- Instances are represented by the *x*, *y* location on the road. We also know when a car changes lane, but not what lane to.
- True clusters are very elongated and horizontally aligned with the lane central lines
- Regular k-means performs poorly on this problem instead finding spherical clusters.

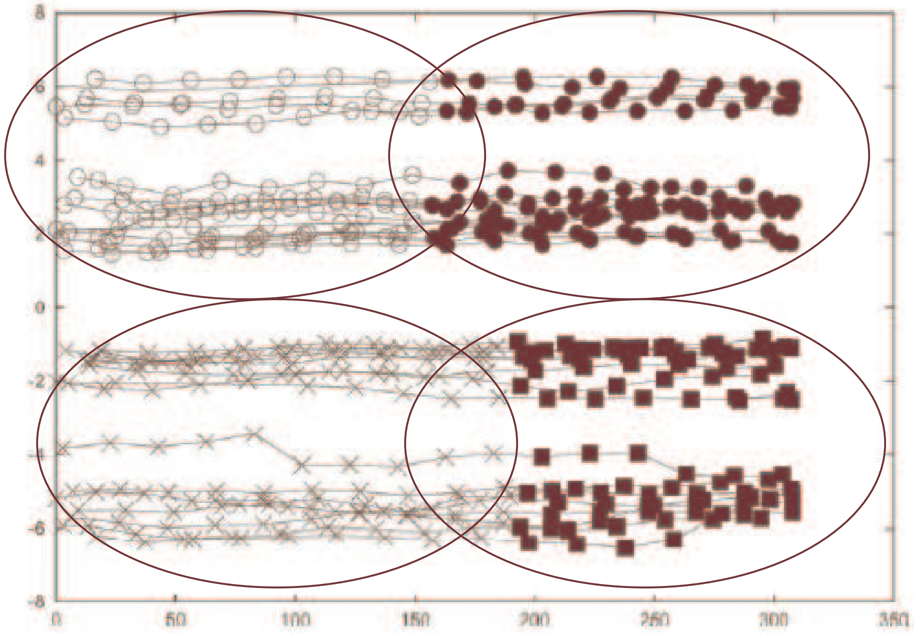


Figure 9. *k*-means output for data set 6, $k = 4$, with nearest clusters marked with different symbols.

Outline

- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

A Quick Summary

- Benefits
  - Increase accuracy when measured on extrinsic labels
  - Obtain clusterings with desired properties
  - Limited results for increasing algorithm run-time (agglomerative hierarchical clustering only)
- Problems
  - Feasibility issues, can easily over-constrain problem
  - Not all constraint sets improve accuracy

The Feasibility Problem

- We’ve seen that constraints are useful …
- But is there a catch?
- We are now trying to find a clustering under all sorts of constraints

Feasibility Problem

Given a set of data points *S*, a set of *ML* and *CL* constraints,

a lower (K_L) and upper bound (K_u) on the number of clusters,

is there **at least one** single set partition of *S* into *k* blocks, $K_U \geq \text{k} \geq K_L$

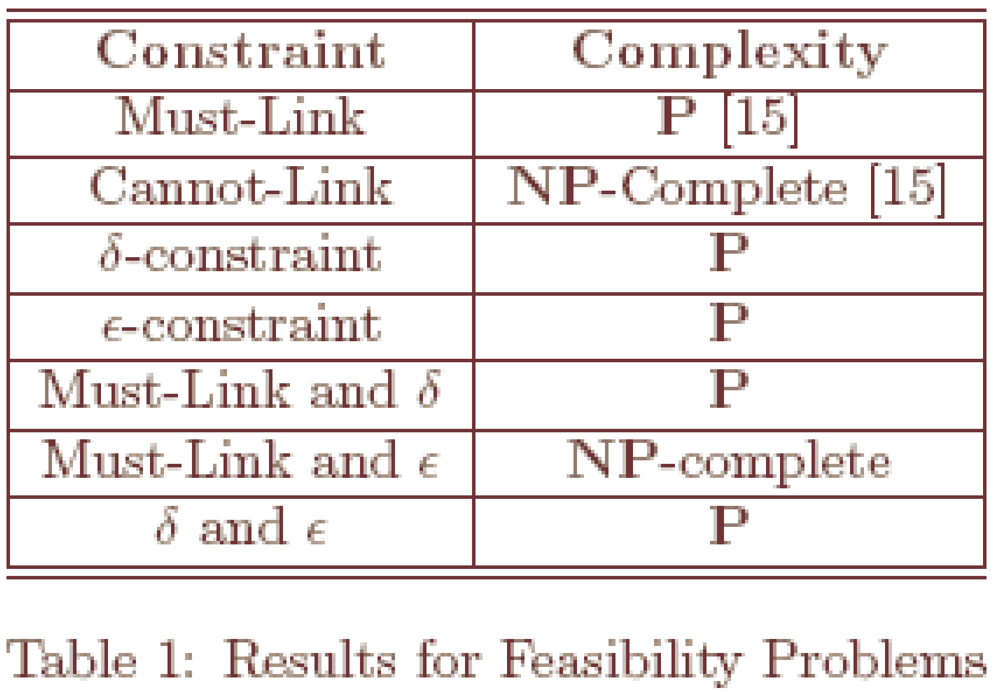
such that no constraints are violated?

i.e. CL(a,b), CL(b,c), CL(a,c), k=2?

Investigating the Feasibility Problem and Consequences?

- For a constraint type or combination:
  - P :construct a polynomial time algorithm
  - NP-complete : reduce from known NP-complete problem
- If the feasibility problem is in P then we can:
  - Use the algorithms to check if a single feasible solution exists before we even apply K-Means
  - Add feasibility checking as a step in K-Means.
- If feasibility problem is NP-complete then:
  - If we try to find a feasible solution at each iteration of K-Means, could take a long time as problem is intractable.

Summary of Feasibility Complexity Results



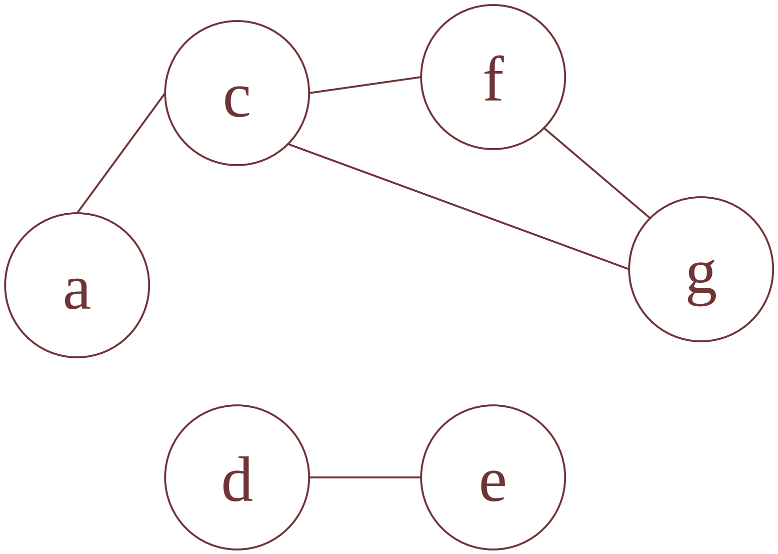
Constraint	Complexity
Must-Link	**P** [15]
Cannot-Link	**NP**-Complete [15]
δ-constraint	**P**
ϵ-constraint	**P**
Must-Link and δ	**P**
Must-Link and ϵ	**NP**-complete
δ and ϵ	**P**

Table 1: Results for Feasibility Problems

Cannot Link Example

Instances a thru z

Constraints: CL(a,c), CL(d,e), CL(f,g), CL(c,g), CL(c,f)



Graph K-coloring problem

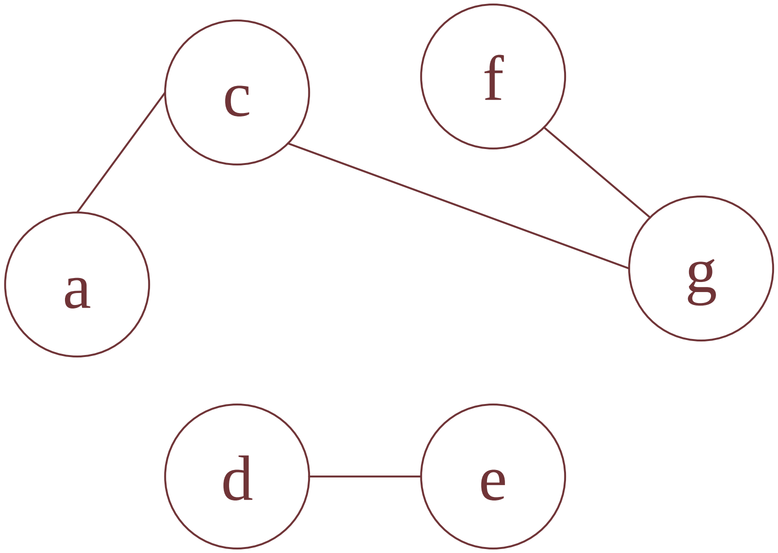
Graph K-coloring problem is intractable for all values of K≥3

See [Davidson and Ravi ’05] for polynomial reduction from graph K-coloring problem.

Must Link Example

Instances a …z

ML(a,c), ML(d,e), ML(f,g), ML(c,g)



M1={a,c,f,g}
M2={d,e}

Let r be the size of the transitive closure (i.e. r=2 above), the number of connected components

Infeasible if k > (n-|TC|)– r
> 26-6 – 2

i.e., can't have too many clusters

New Results

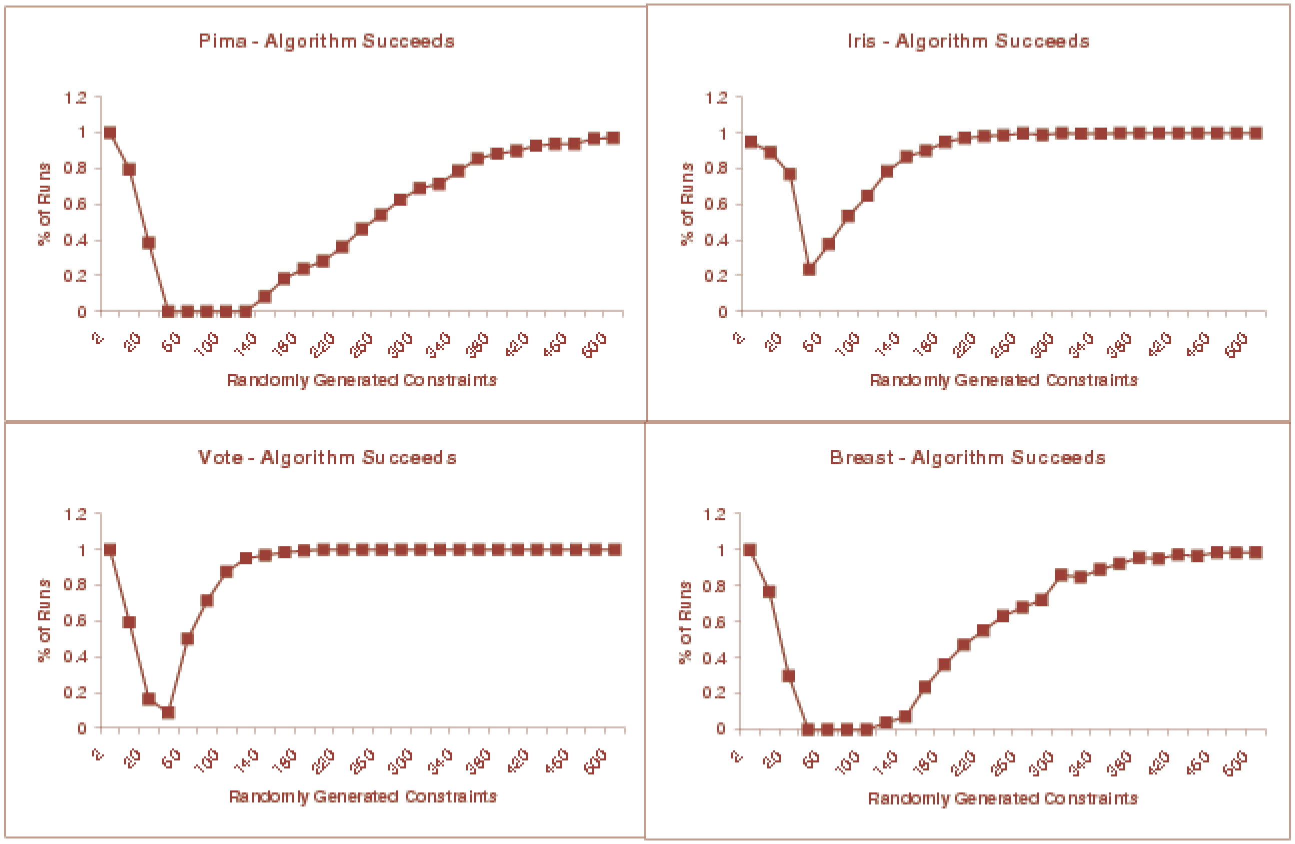
- Feasibility Problem for Disjunctions of ML and CL constraints are intractable
- But Feasibility Problem for Choice sets of ML and CL constraints are easy.
  - $\mathrm{ML}(\mathbf{x}, y_1) \vee \mathrm{ML}(\mathbf{x}, y_2) \ldots \vee \mathrm{ML}(\mathbf{x},y_n)$
  - i.e. x must-be linked with one of the y's.

Is **Over-constraining** Really a Problem

- Wait! You said clustering under cannot link constraints was intractable.
- Worst case results say that there is one at least one “hard” problem instance so pessimistically we say the entire problem is hard.
- But when and how often does **over-constraining** become a problem.
- Set k = # extrinsic clusters
- Randomly generated constraints by choosing two instances
- Run COP-k-means

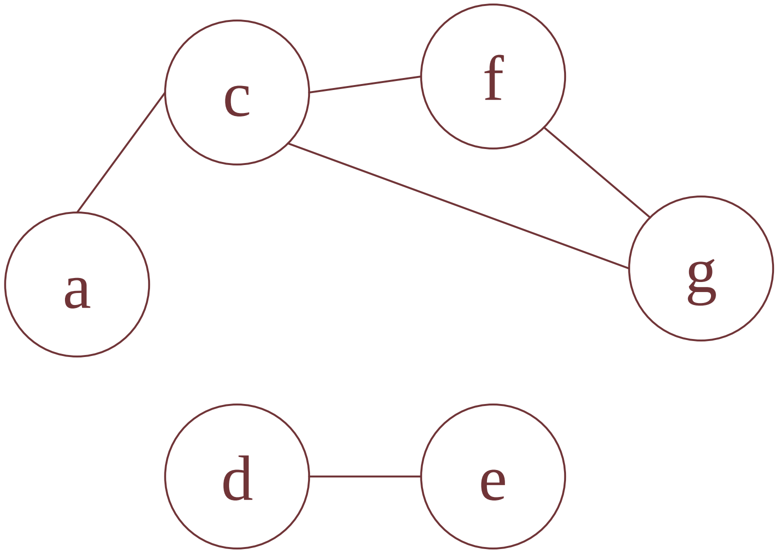
Experimental Results

Figure 3: Graph of the proportion of times from 500 independent trials the algorithm in figure 2 gets stuck for various number of randomly chosen ML and CL constraints, k = number of instrinsic classes: Iris (3), Pima (2), Breast (2) and Vote (2).



Some Theoretical Results To Identify Easy Constraint Sets

[Davidson, Ravi AAAI ’06]



Identify sufficient conditions where coloring is easy and hence algorithms like COP-k-means will always converge if a feasible solution exists.

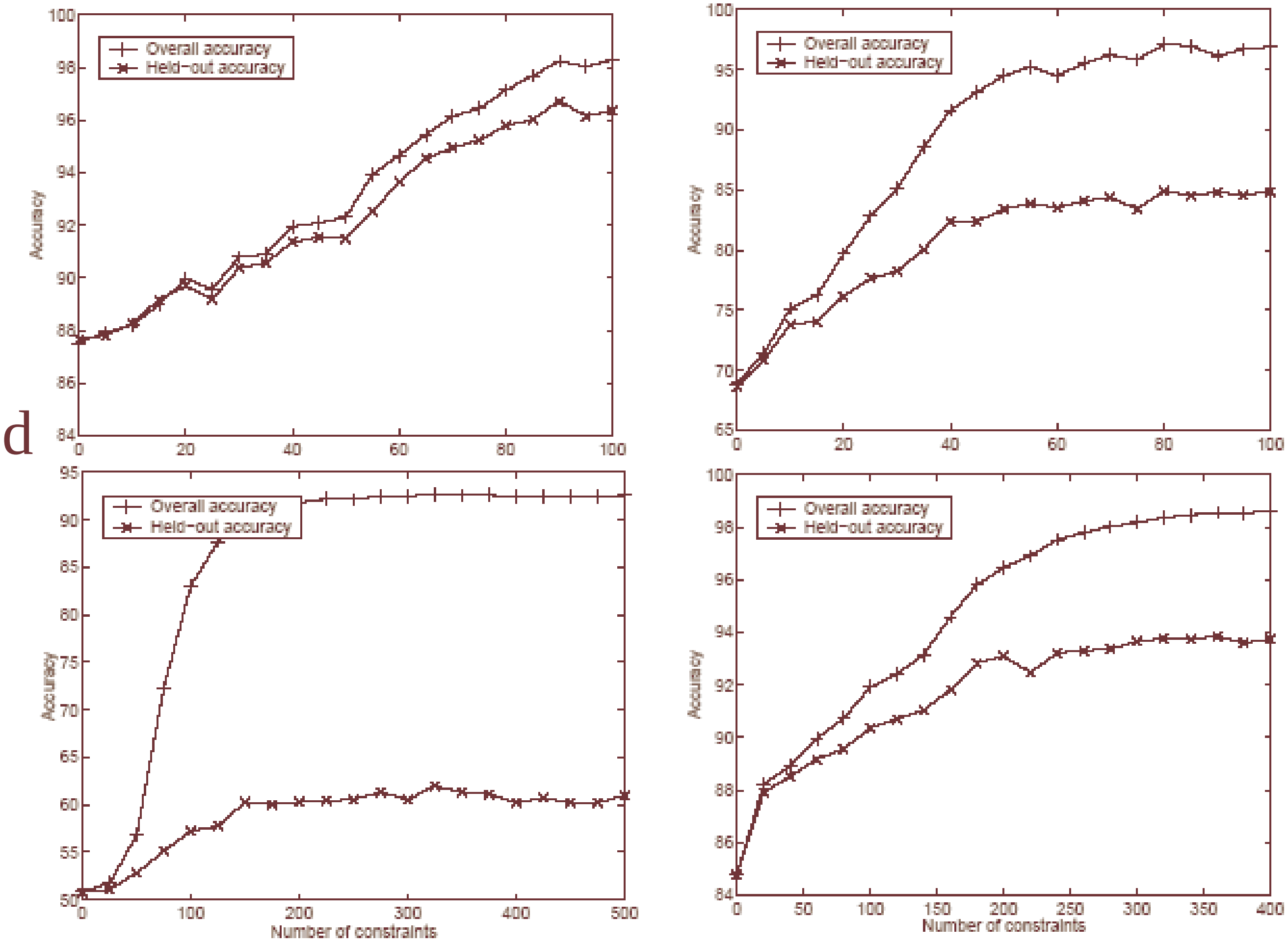
a) If k ≥ maxDegree(CL-Graph) + 1

b) If k ≥ Q-Induct(CL-Graph) + 1
Q-inductiveness of a graph: Ordering of instances and assigned integer values so that at most *Q* edges point down-stream.

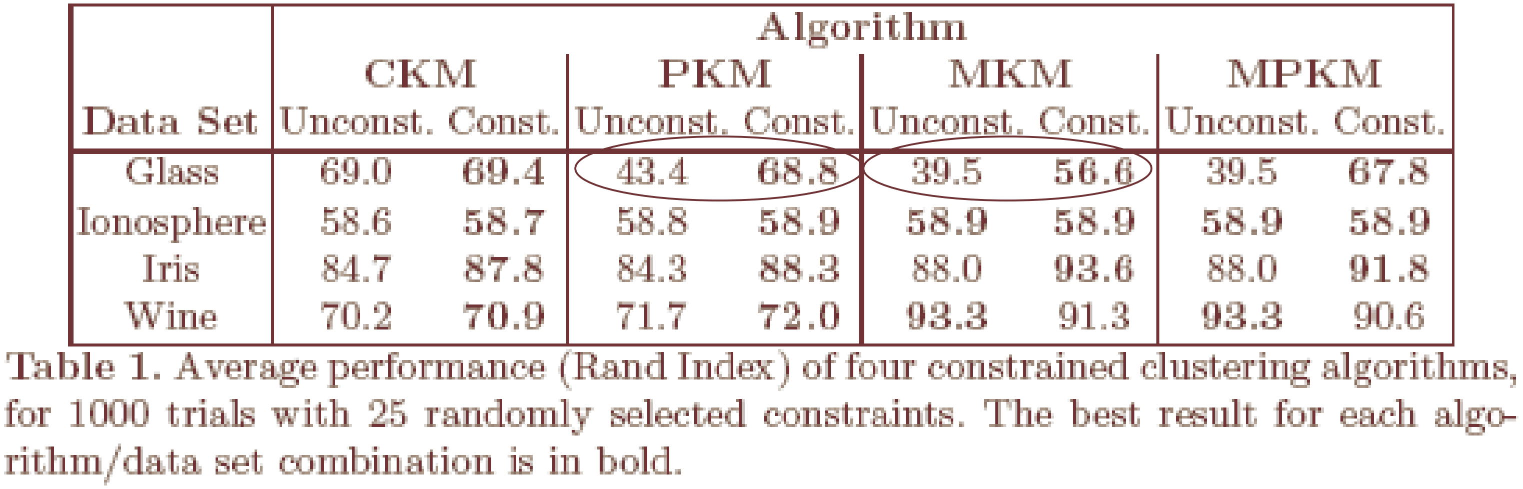
Can Constraints Adversely Effect Performance?

Many people (including ourselves) Reported averaged performance

[Wagstaff ’02]

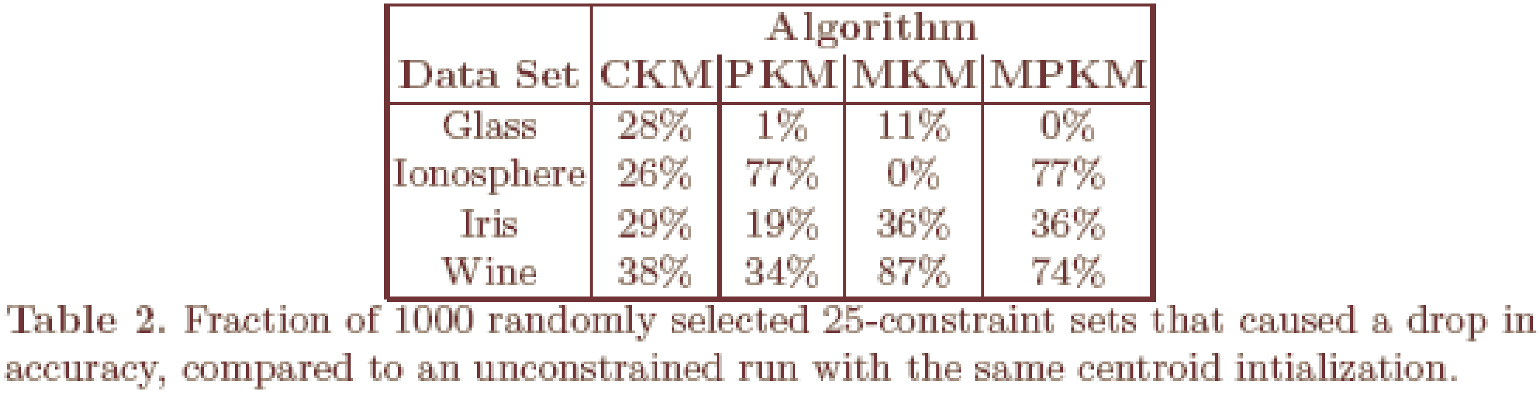


However Averaging Masks That Some Constraint Sets Have Adverse Effects



	Algorithm							
	CKM		PKM		MKM		MPKM	
Data Set	Unconst.	Const.	Unconst.	Const.	Unconst.	Const.	Unconst.	Const.
Glass	69.0	69.4	43.4	68.8	39.5	56.6	39.5	67.8
Ionosphere	58.6	58.7	58.8	58.9	58.9	58.9	58.9	58.9
Iris	84.7	87.8	84.3	88.3	88.0	93.6	88.0	91.8
Wine	70.2	70.9	71.7	72.0	93.3	91.3	93.3	90.6

Table 1. Average performance (Rand Index) of four constrained clustering algorithms, for 1000 trials with 25 randomly selected constraints. The best result for each algorithm/data set combination is in bold.

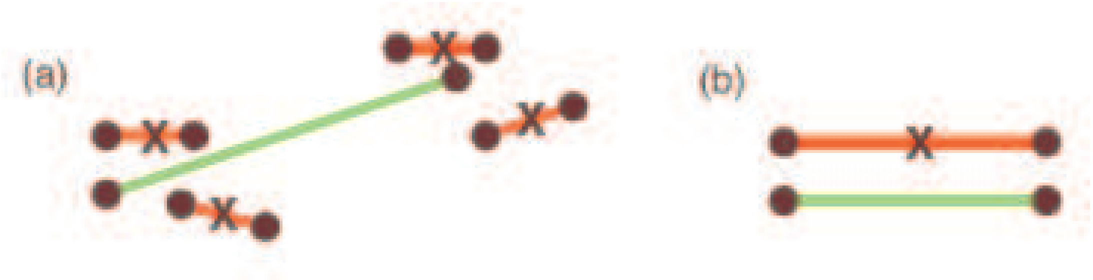


	Algorithm			
Data Set	CKM	PKM	MKM	MPKM
Glass	28%	1%	11%	0%
Ionosphere	26%	77%	0%	77%
Iris	29%	19%	36%	36%
Wine	38%	34%	87%	74%

Table 2. Fraction of 1000 randomly selected 25-constraint sets that caused a drop in accuracy, compared to an unconstrained run with the same centroid intialization.

Identifying Useful Constraint Sets: Informativeness and Coherence

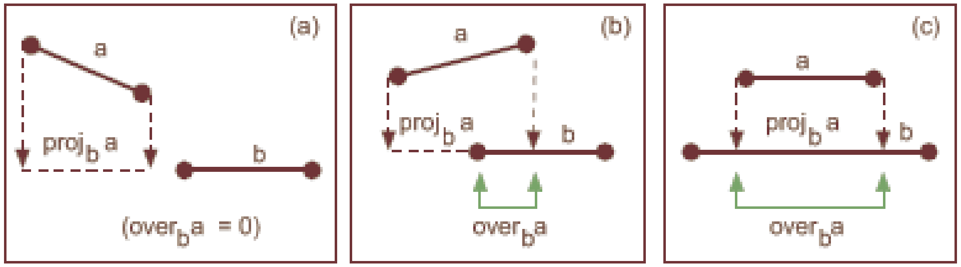
[Davidson, Wagstaff, Basu ’06]



Informativeness

Coherence

$$\mathcal{I}_A(C) = \frac{1}{|C|}\left[\sum_{c\in C} unsat(c, P_A)\right]$$



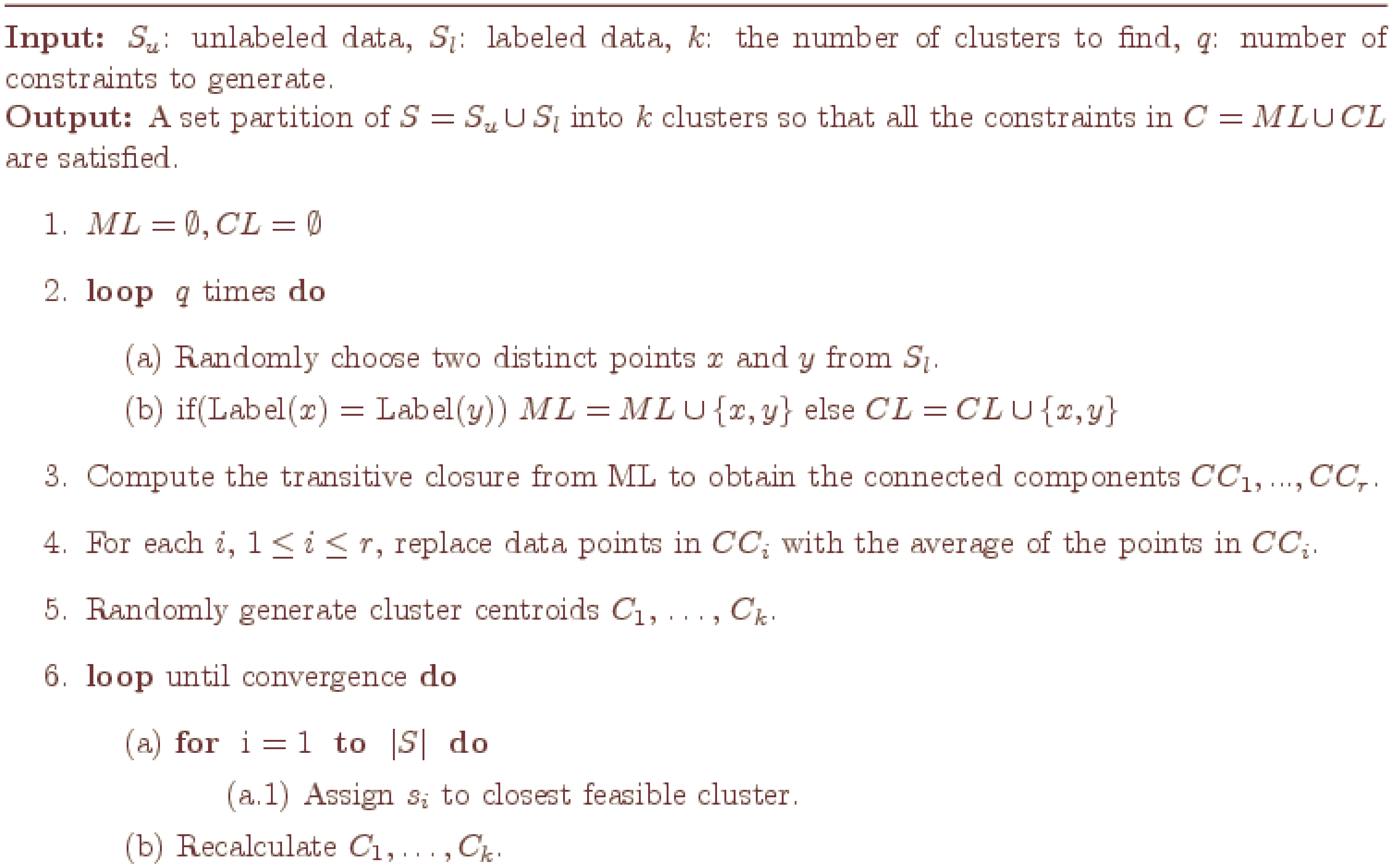
Outline

• Introduction and Motivation	[Ian]
• Uses of constraints	[Sugato]
• Real-world examples	[Sugato]
• Benefits and problems of using constraints	[Ian]
• Algorithms for constrained clustering	
• Enforcing constraints	[Ian]
• Hierarchical	[Ian]
• Learning distances	[Sugato]
• Initializing and pre-processing	[Sugato]
• Graph-based	[Sugato]

Enforcing Constraints

- Constraints are strong background information that should be satisfied.
- Two options
  - Satisfy all constraints, but we will run into infeasibility problems
  - Satisfy as many constraints as possible, but working out largest subset of constraints is also intractable (largest-color problem)

COP-k-Means – Nearest-“Feasible”- Centroid Idea

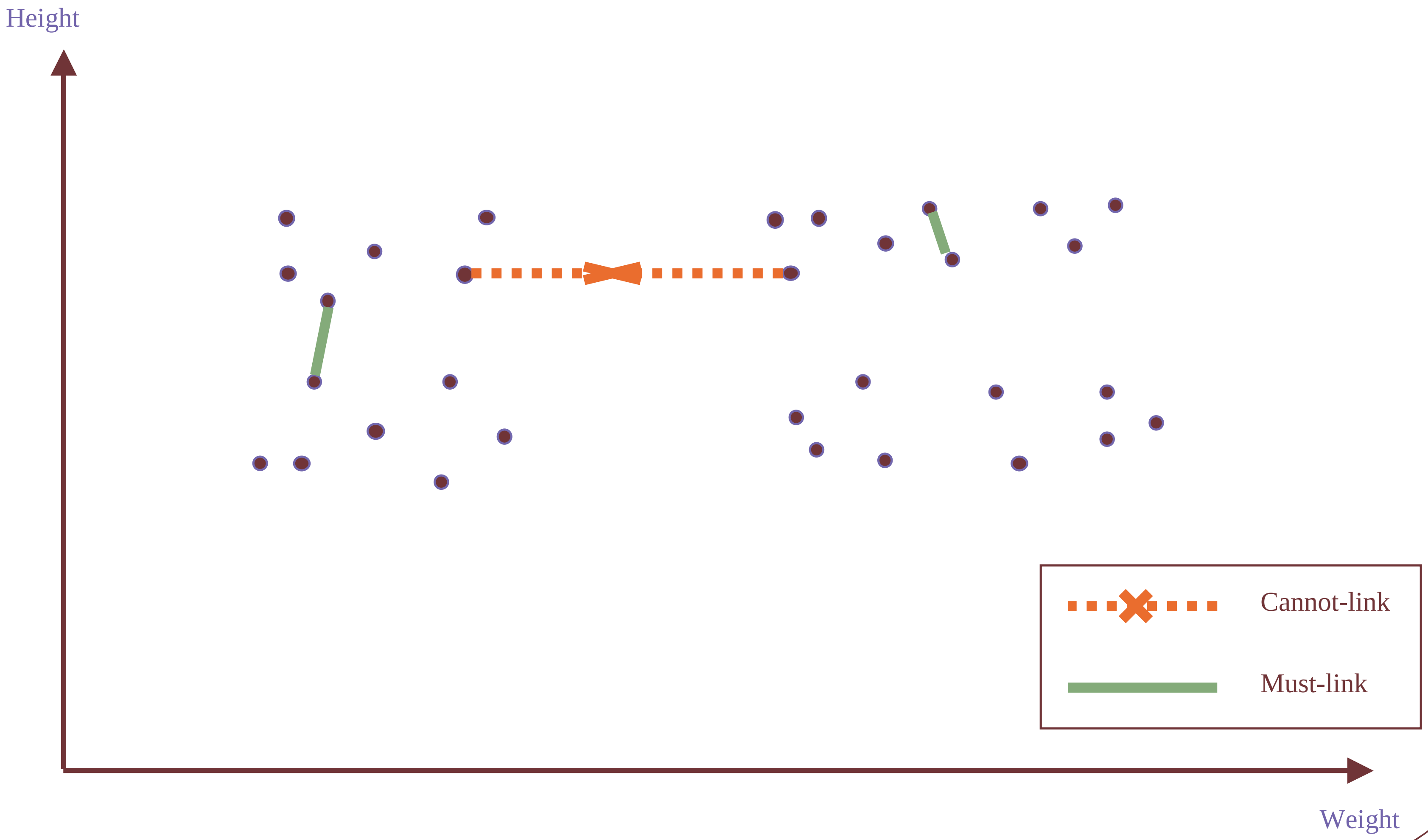


Input: S_u: unlabeled data, S_l: labeled data, k: the number of clusters to find, q: number of constraints to generate.

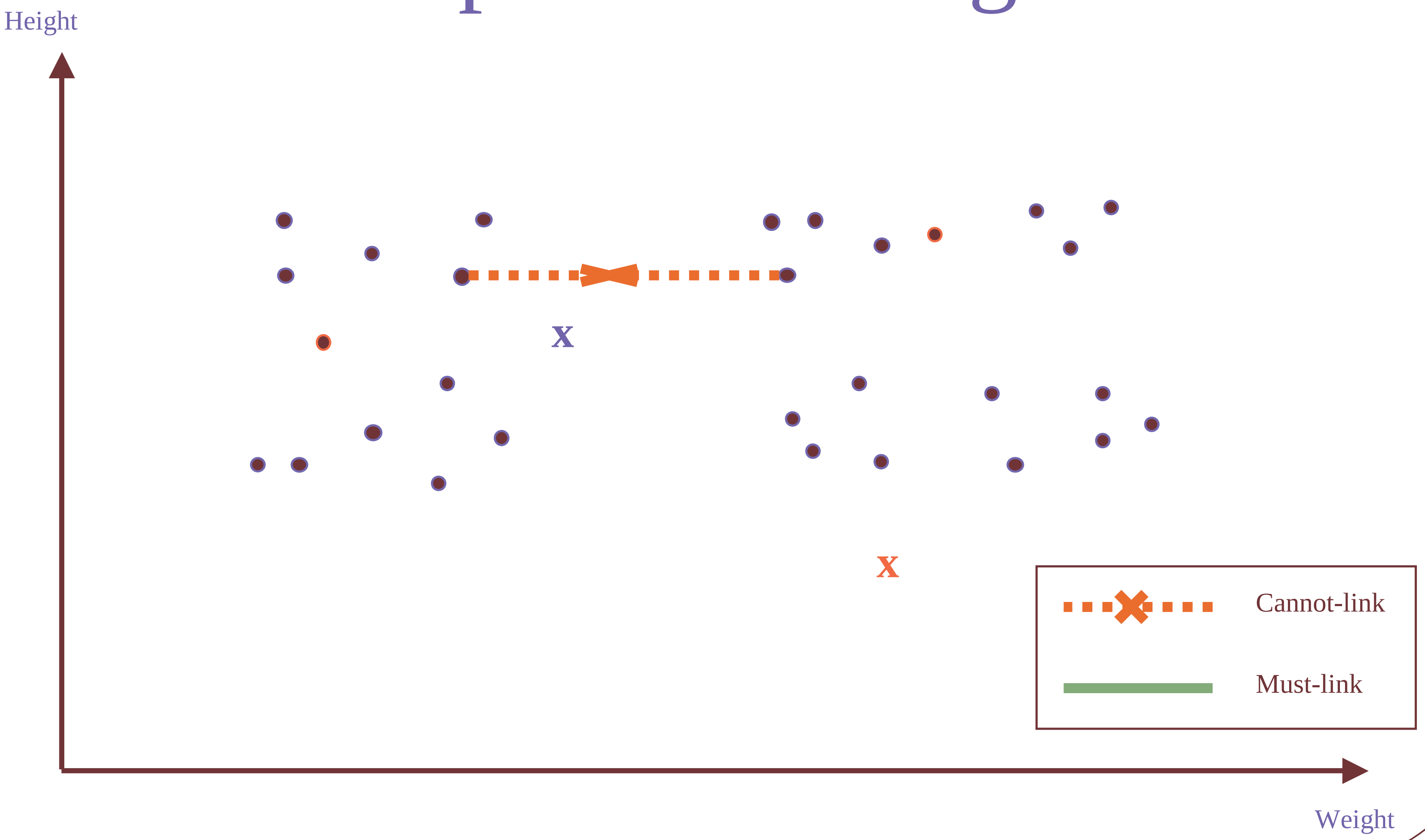
Output: A set partition of $S = S_u \cup S_l$ into k clusters so that all the constraints in $C = ML \cup CL$ are satisfied.

```
1. $ML = \emptyset, CL = \emptyset$
2. loop $q$ times do
    (a) Randomly choose two distinct points $x$ and $y$ from $S_l$.
    (b) if(Label($x$) = Label($y$)) $ML = ML \cup \{x,y\}$ else $CL = CL \cup \{x,y\}$
3. Compute the transitive closure from ML to obtain the connected components $CC_1, ..., CC_r$.
4. For each $i$, $1 \le i \le r$, replace data points in $CC_i$ with the average of the points in $CC_i$.
5. Randomly generate cluster centroids $C_1, \ldots, C_k$.
6. loop until convergence do
    (a) for i = 1 to $|S|$ do
        (a.1) Assign $s_i$ to closest feasible cluster.
    (b) Recalculate $C_1, \ldots, C_k$.
```

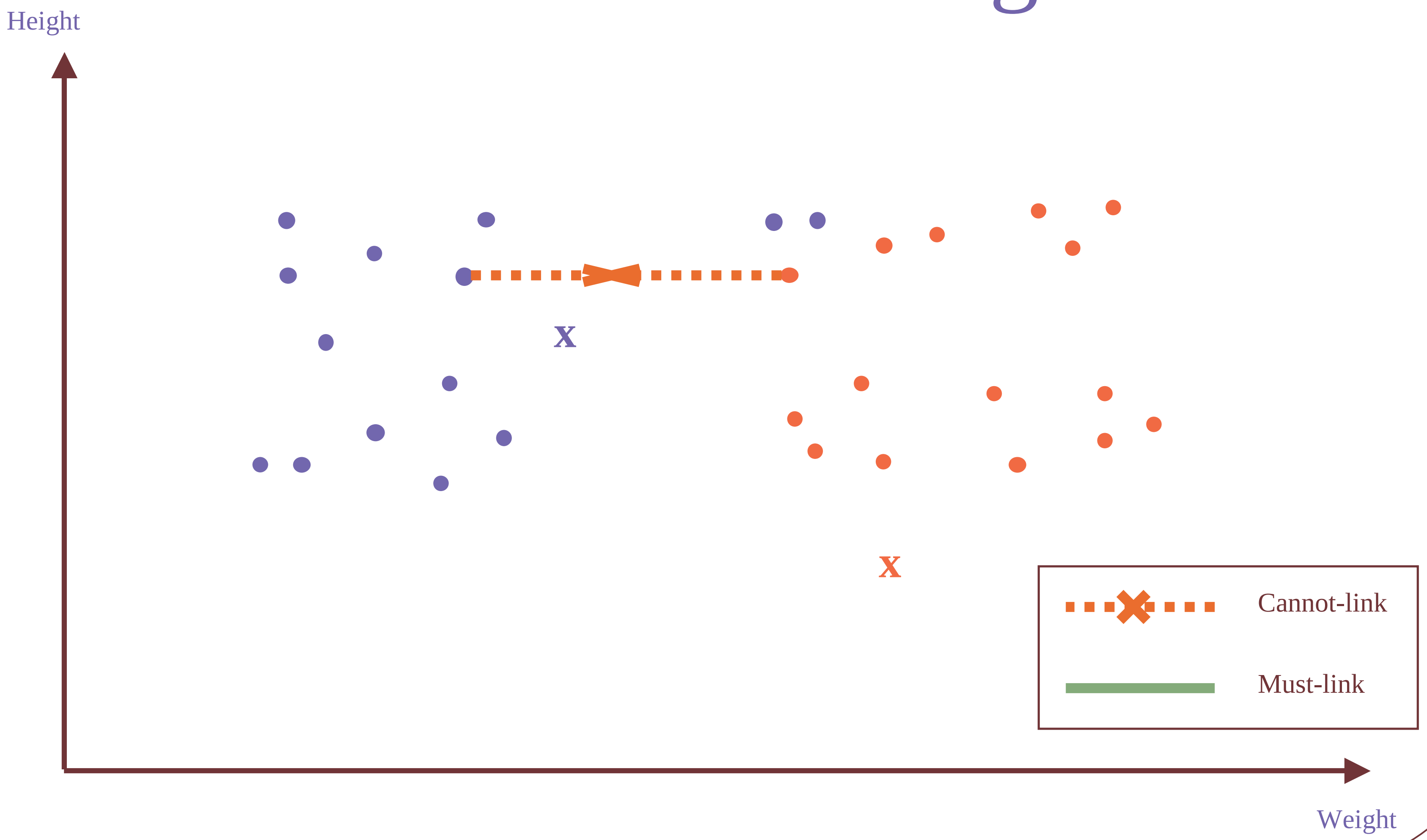
Example: COP-K-Means - 1



Example: COP-K-Means – 2 ML points Averaged



Example: COP-K-Means – 3
Nearest-Feasible-Assignment



Trying To Minimize VQE and Satisfy As Many Constraints As Possible

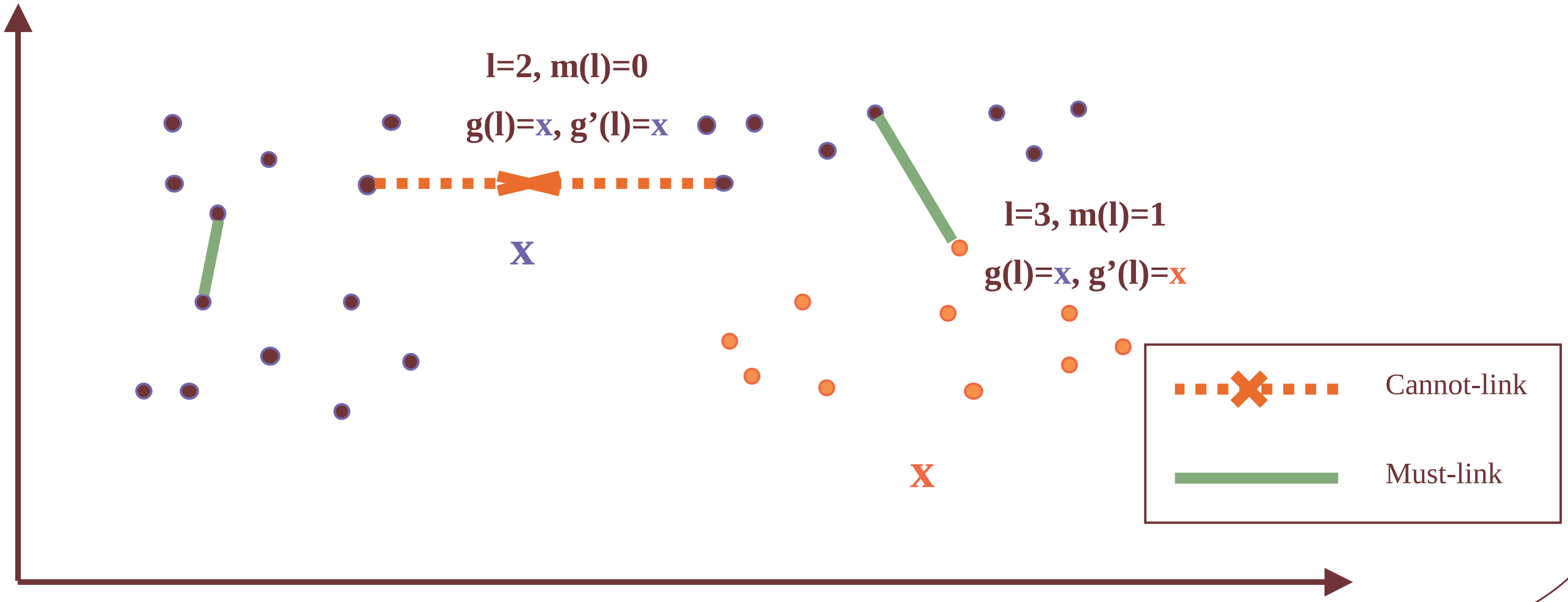
- Can't rely on expecting that I can satisfy all constraints at each iteration.
- Change aim of K-Means from:
  - Find a solution satisfying all the constraints and minimizing VQE

TO

  - Find a solution satisfying most of the constraints (penalized if a constraint is violated) and minimizing VQE
- Two tricks
  - Need to express penalty term in same units as VQE/distortion
  - Need to rederive K-Means (as a gradient descent algorithm) from first principles.

An Approximation Algorithm – Notation

g(l), g'(l) and m(l) refer to the l^{th} constraint
g(l) : assigned cluster for first instance in constraint
g'(l) : assigned cluster for second instance in constraint
m(l) = 1 for must link, m(l) = 0 for cannot link



New Differentiable Objective Function

Satisfying a constraint may increase distortion

Trade-off between satisfying constraints and distortion requires measurement in the same units

$$(5.5) \qquad CVQE_j = \frac{1}{2}\sum_{s_i \in Q_j} T_{j,1} + \frac{1}{2}\sum_{l=1,g(l)=j}^{s+r} (T_{j,2} \times T_{j,3})$$

where

$$T_{j,1} = (C_j - s_i)^2$$

$$\left.\begin{aligned} T_{j,2} &= \mathbf{k}_1\left[(C_j - C_{g'(l)})^2 \neg \Delta(g'(l), g(l))\right]^{m_l} \\ T_{j,3} &= \mathbf{k}_2\left[(C_j - C_{h(g'(l))})^2 \Delta(g(l), g'(l))\right]^{1-m_l} \end{aligned}\right\}.$$

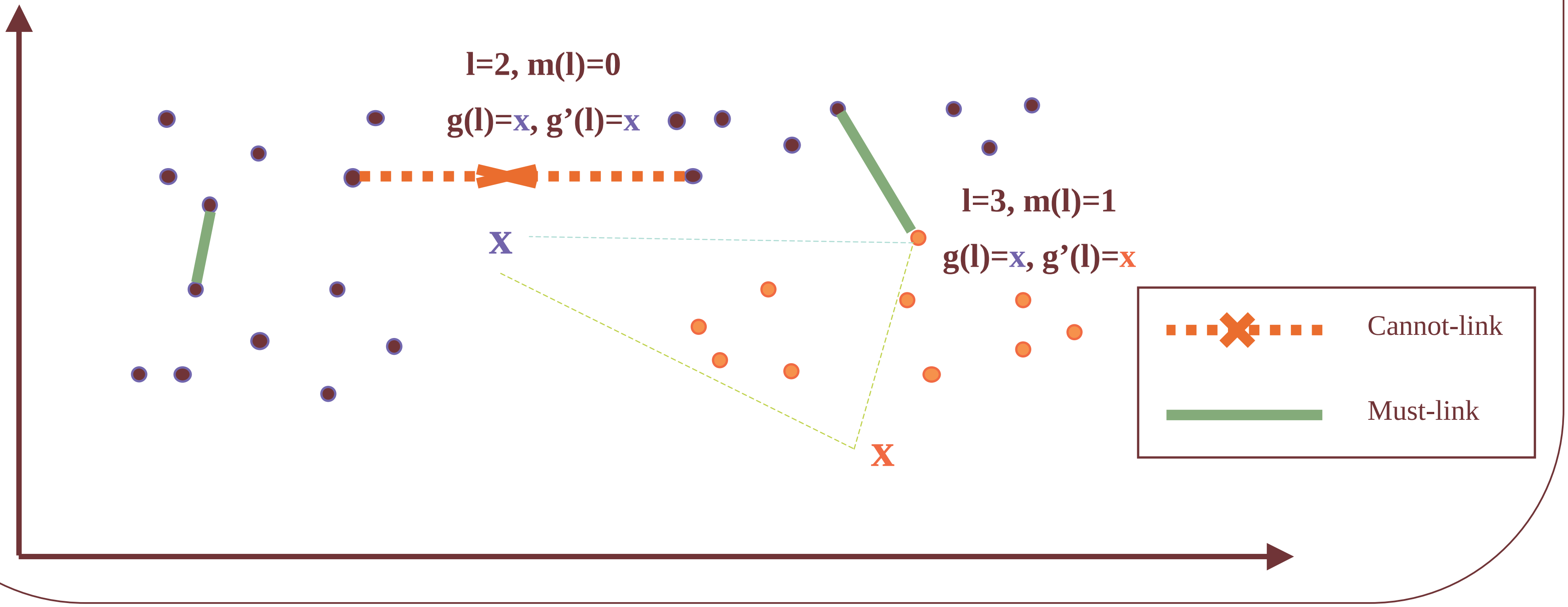
Only one is non-zero per constraint violation

If ML violated add distance between clusters

If CL violated add distance between cluster and nearest cluster

Visualizing the Penalties

Either satisfy the constraint, or
Assign to the "nearest" centroid but with a penalty



Constrained K-Means Algorithm

Algorithm aims to minimize CVQE and has a formal derivation

Randomly assign each instance to a cluster.

1. C_j = *Average of points assigned to j*

   + Centroids of points that ***should be*** *assigned to j*

   + Nearest Centroids to points ***that should not to be*** *assigned to j*

2. NN assignment for each instance using new distance

   Assign x to C_j iff $argmin_j \, CVQE(x,C_j)$

Goto 1 until $\Delta CVQE$ is small

Must Link Penalties

Cannot Link Penalties

Outline

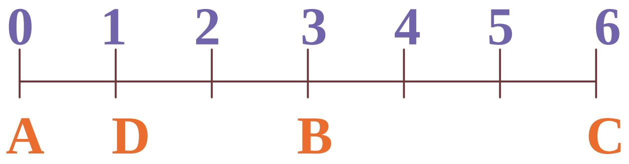
- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Hierarchical Clustering

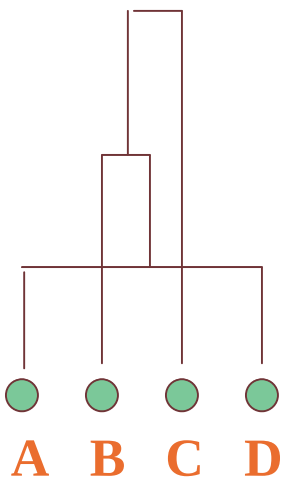
Agglomerative Hierarchical Clustering

1. Initially, every instance is in its own cluster
2. Compute similarities between each cluster
3. Merge two most **similar** clusters into one.
4. Goto 2

Time Complexity $O(n^2)$

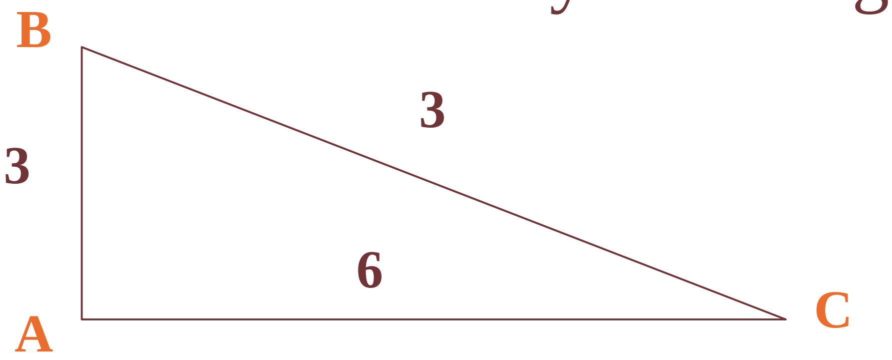


$$D = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 3 & 6 & 1 \\ B & 3 & 0 & 3 & 2 \\ C & 6 & 3 & 0 & 5 \\ D & 1 & 2 & 5 & 0 \end{array}$$



Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) - 1

- Metric spaces. Only changing the distance matrix not the distance function.
- But we must satisfy the triangle inequality



$d(x,y) \leq d(x,z) + d(z,y)$

$d(x,y) \geq | d(x,z) - d(z,y) |$

- If inequality did not hold then shortest distance between two points wouldn’t be a line.

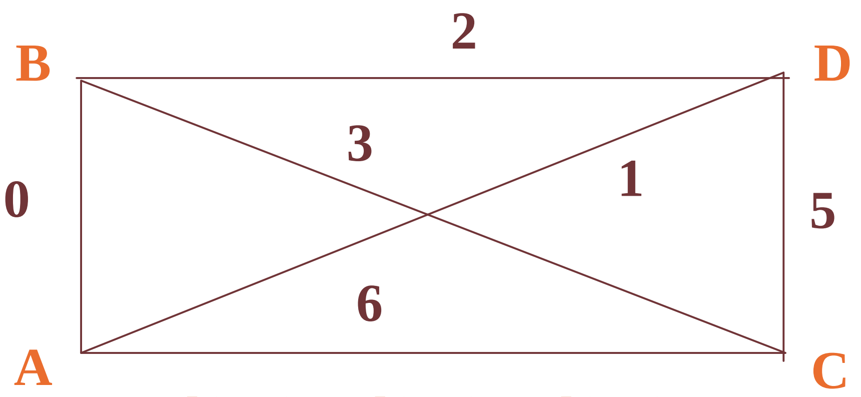
Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) - 2

$$D = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 0 & 6 & 1 \\ B & 0 & 0 & 3 & 2 \\ C & 6 & 3 & 0 & 5 \\ D & 1 & 2 & 5 & 0 \end{array}$$

Causes Violation

ML(A,B)

CL(A,D)



d(x,y) ≤ d(x,z) + d(z,y)

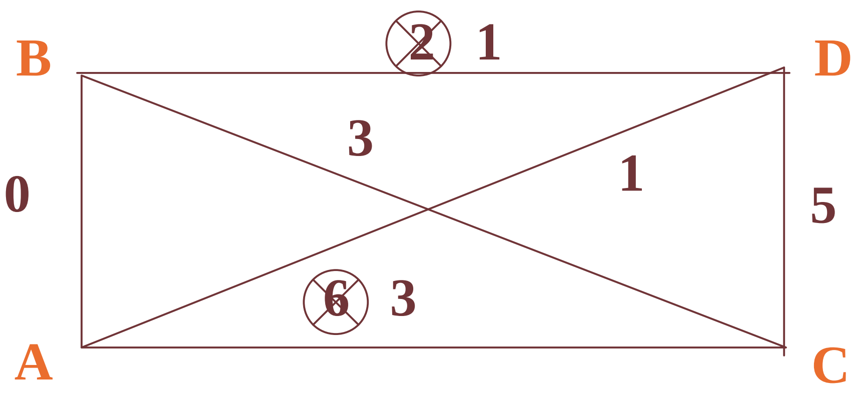
d(x,y) ≥ | d(x,z) – d(z,y) |

Algorithm

- 1): Change ML distance instance entries in D to 0
- 2): Calculate D’ from D using all pairwise shortest path algorithms, takes $O(n^3)$
- 3): D’’ = D’ Except Change CL distance entries to be max(D)+1

Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) – 3

STEP 2



$$D' = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 0 & 3 & 1 \\ B & 0 & 0 & 3 & 1 \\ C & 3 & 3 & 0 & 5 \\ D & 1 & 1 & 5 & 0 \end{array}$$

$$d(x,y) \leq d(x,z) + d(z,y)$$

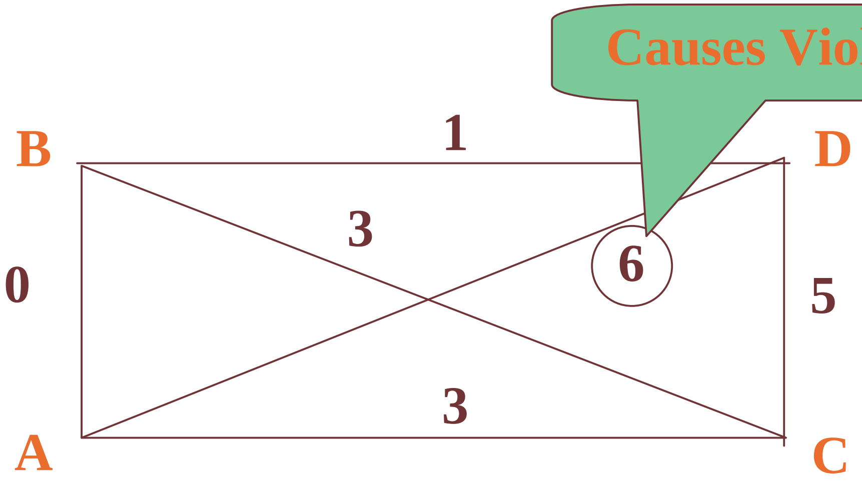
$$d(x,y) \geq | d(x,z) - d(z,y) |$$

Algorithm

- 1): Change ML distance instance entries in D to 0
- 2): Calculate D' from D using all pairwise shortest path algorithms, takes *O(n³)*
- 3): D'' = D' Except Change CL distance entries to be max(D)+1

Modify the Distance Matrix (D) To Satisfy Instance Level Constraints (KKM02) - 4

STEP 3



$$D'' = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 0 & 3 & 6 \\ B & 0 & 0 & 3 & 1 \\ C & 3 & 3 & 0 & 5 \\ D & 6 & 1 & 5 & 0 \end{array}$$

But Because of entailment property of CL we “maintain” the triangle inequality

Join(A,B)

Can’t Join((A,B),D) instead Join((A,B),C) and then stop

Indirectly made d(B,D) and d(A,C) >> 6 and make inequality indirectly hold.

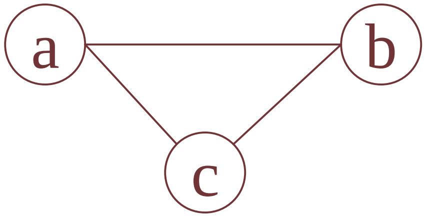
Feasibility, Dead-ends and Speeding Up Agglomerative Clustering

Feasibility Problem

Instance: Given a set S of points, a (symmetric) distance function d(x,y)≥0 ∀x,y and a collection of *C* constraints.

Problem: Can *S* be partitioned into **at least one** single subsets (clusters) so that all constraints are satisfied?

CL(a,b),
CL(b,c),
CL(a,c)
(k=3, k=2, k=1)?



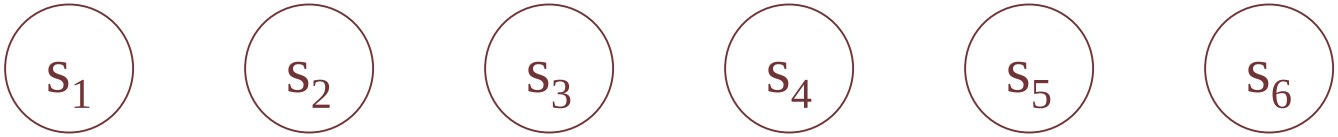
For fixed *k* equivalent to graph coloring so NP-complete

Feasibility Results

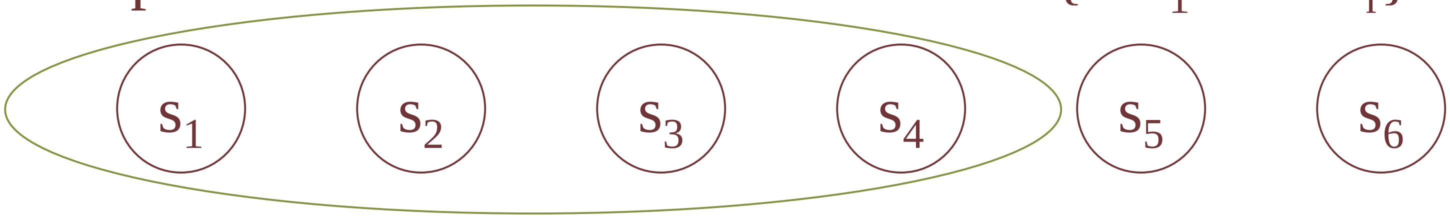
Constraint	**Given *k***	**Unspecified *k***
ML	**P** [SDM05]	**P** [PKDD05]
CL	**NP-complete** [SDM05]	**P** [PKDD05]
δ	**P** [SDM05]	**P** [PKDD05]
ε	**P** [SDM05]	**P** [PKDD05]
ML and ε	**NP-complete** [SDM05]	**P** [PKDD05]
ML and δ	**P** [SDM05]	**P** [PKDD05]
δ and ε	**P** [SDM05]	**P** [PKDD05]
ML, CL and ε	**NP-complete** [SDM05]	**NP-complete** [PKDD05]

Feasibility under ML and CL

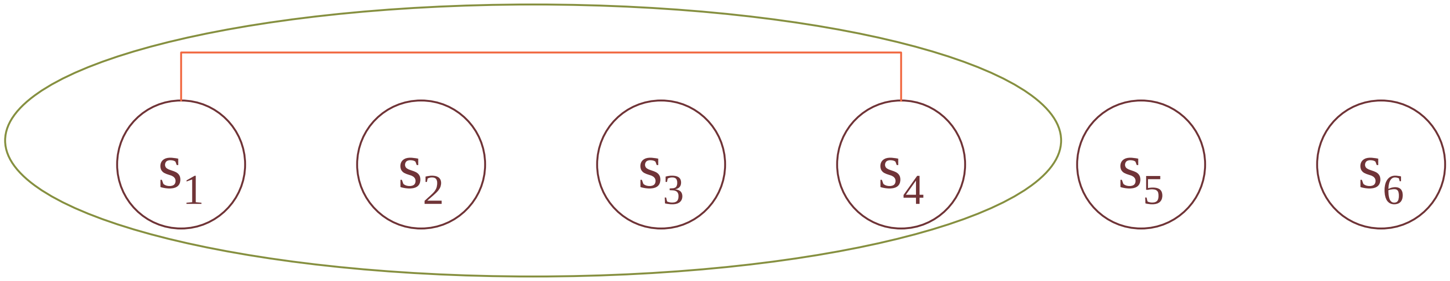
$ML(s_1,s_3)$, $ML(ML(s_2,s_3)$, $ML(s_2,s_4)$, $CL(s_1, s_4)$



Compute the Transitive Closure on $ML=\{CC_1 \ldots CC_r\}$ $O(n+m_{ML})$



Construct Edges {E} between Nodes based on CL: $O(m_{CL})$

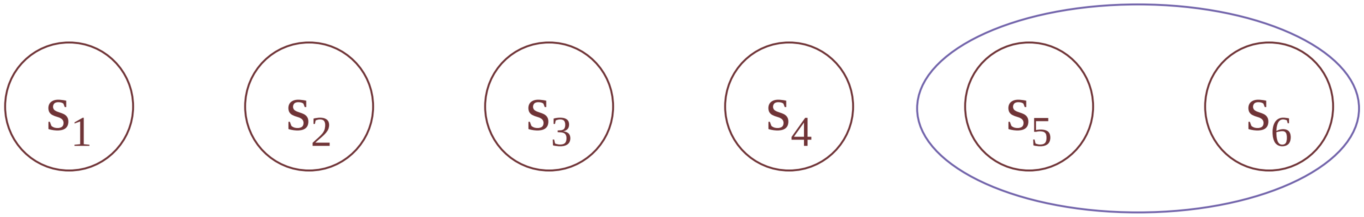


Infeasible: iff $\exists h,k : e_h(s_i, s_j) : s_i, s_j \in CC_k : O(m_{CL})$

Feasibility under ML and ε

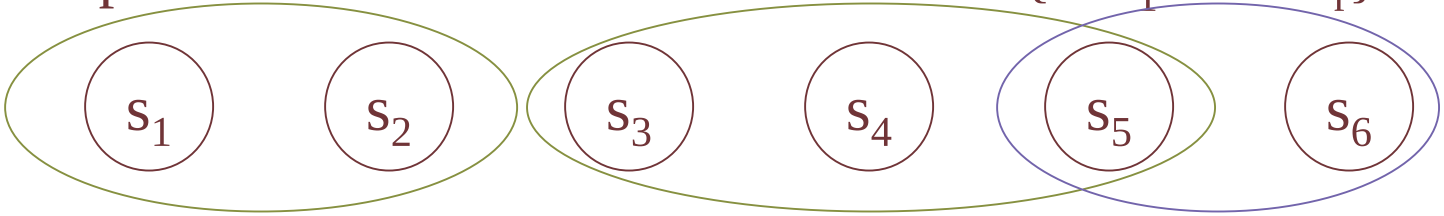
$S' = \{x \in S : x \text{ does } \textbf{not} \text{ have an } \varepsilon \text{ neighbor}\} = \{s_5, s_6\}$

Each of these should be in their own cluster



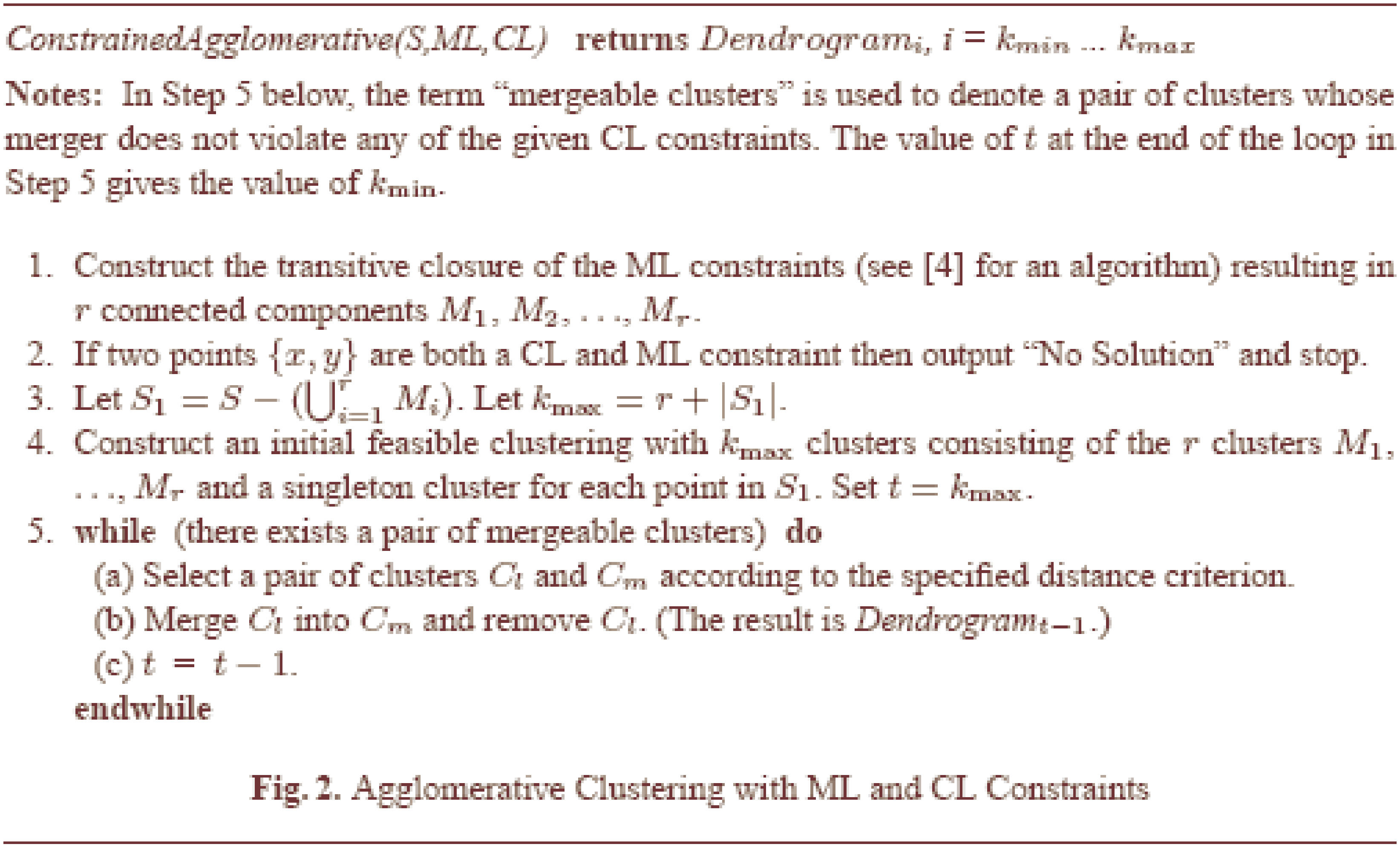
$ML(s_1,s_2)$, $ML(s_3,s_4)$, $ML(s_4,s_5)$

Compute the Transitive Closure on $ML=\{CC_1 \ldots CC_r\}$: $O(n+m)$



Infeasible: iff $\exists i,j : s_i \in CC_j, s_i \in S'$: $O(|S'|)$

An Algorithm for ML and CL Constraints



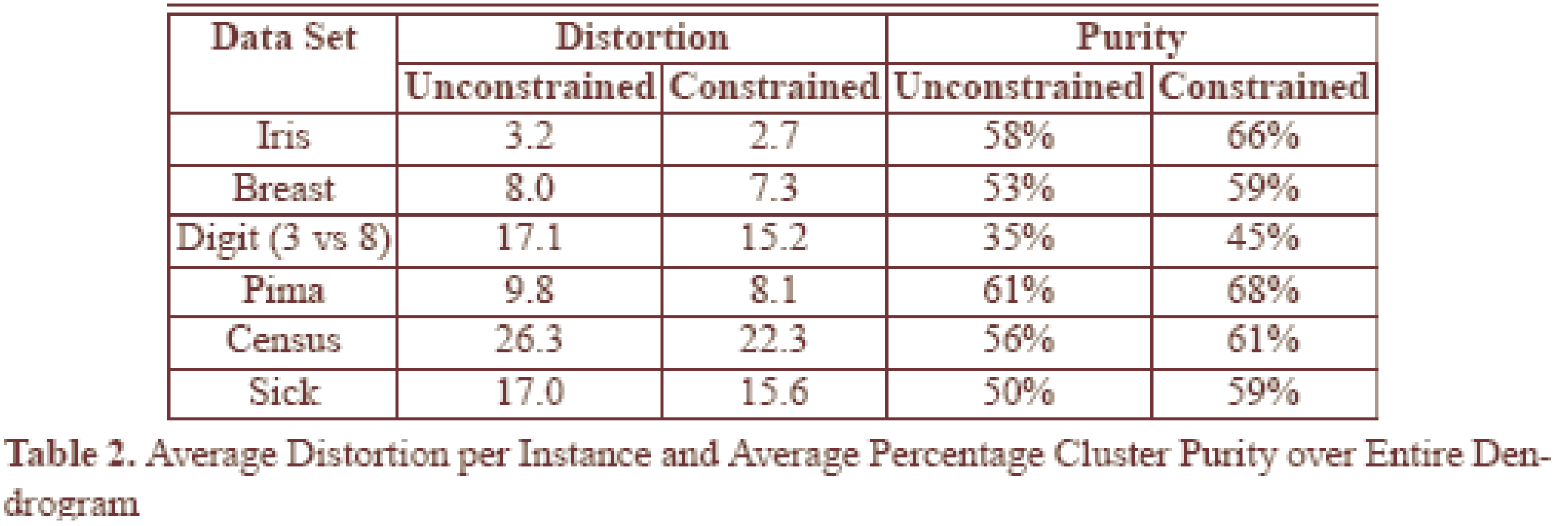
ConstrainedAgglomerative(S,ML,CL) **returns** $Dendrogram_i$, $i = k_{min} \ldots k_{max}$

Notes: In Step 5 below, the term "mergeable clusters" is used to denote a pair of clusters whose merger does not violate any of the given CL constraints. The value of t at the end of the loop in Step 5 gives the value of $k_{\min}$.

```
1. Construct the transitive closure of the ML constraints (see [4] for an algorithm) resulting in
   $r$ connected components $M_1, M_2, \ldots, M_r$.
2. If two points $\{x, y\}$ are both a CL and ML constraint then output "No Solution" and stop.
3. Let $S_1 = S - (\bigcup_{i=1}^{r} M_i)$. Let $k_{\max} = r + |S_1|$.
4. Construct an initial feasible clustering with $k_{\max}$ clusters consisting of the $r$ clusters $M_1$,
   $\ldots$, $M_r$ and a singleton cluster for each point in $S_1$. Set $t = k_{\max}$.
5. while (there exists a pair of mergeable clusters) do
     (a) Select a pair of clusters $C_l$ and $C_m$ according to the specified distance criterion.
     (b) Merge $C_l$ into $C_m$ and remove $C_l$. (The result is $Dendrogram_{t-1}$.)
     (c) $t = t - 1$.
   endwhile
```

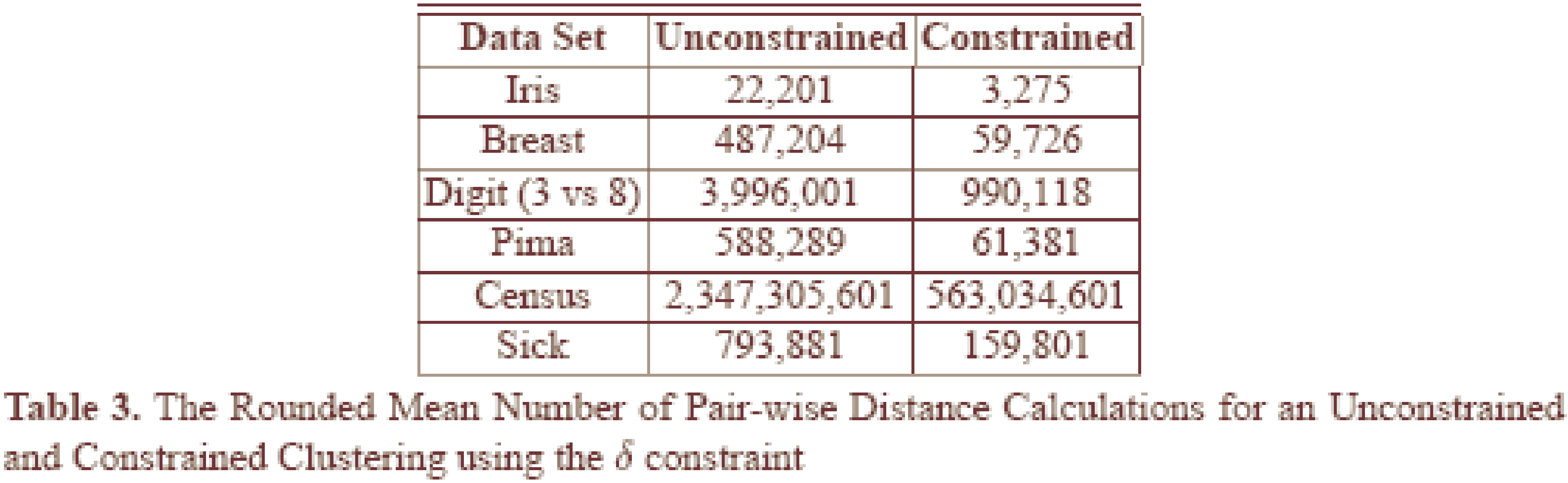
Fig. 2. Agglomerative Clustering with ML and CL Constraints

Empirical Results



Data Set	Distortion		Purity	
	Unconstrained	Constrained	Unconstrained	Constrained
Iris	3.2	2.7	58%	66%
Breast	8.0	7.3	53%	59%
Digit (3 vs 8)	17.1	15.2	35%	45%
Pima	9.8	8.1	61%	68%
Census	26.3	22.3	56%	61%
Sick	17.0	15.6	50%	59%

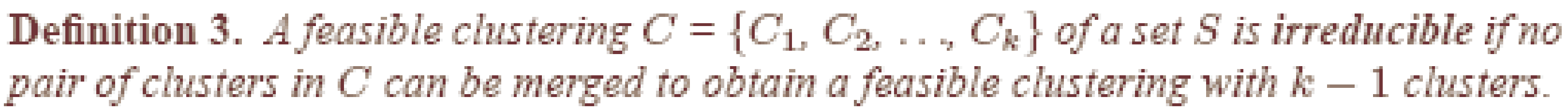
Table 2. Average Distortion per Instance and Average Percentage Cluster Purity over Entire Dendrogram



Data Set	Unconstrained	Constrained
Iris	22,201	3,275
Breast	487,204	59,726
Digit (3 vs 8)	3,996,001	990,118
Pima	588,289	61,381
Census	2,347,305,601	563,034,601
Sick	793,881	159,801

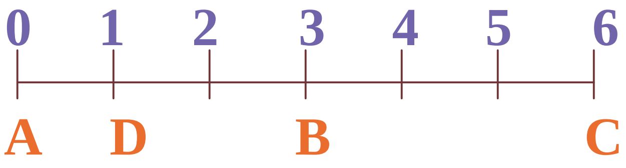
Table 3. The Rounded Mean Number of Pair-wise Distance Calculations for an Unconstrained and Constrained Clustering using the δ constraint

Dead-end Clusterings



Definition 3. *A feasible clustering $C = \{C_1, C_2, \ldots, C_k\}$ of a set S is irreducible if no pair of clusters in C can be merged to obtain a feasible clustering with $k - 1$ clusters.*

A *k* cluster clustering is a dead-end if it is irreducible, even though other feasible clusterings with <*k* clusters exist



$$D = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 3 & 6 & 1 \\ B & 3 & 0 & 3 & 2 \\ C & 6 & 3 & 0 & 5 \\ D & 1 & 2 & 5 & 0 \end{array}$$

Constraints CL(A,B) CL(A,C)

Join(A,D) Can’t go any further – Deadend

Even Though Join(B,C), Join(A,D) is possible

Why Are Dead-Ends a Problem?

- Theorem (in technical report)
  - Let $k_{min} < k_{max}$, then if there is a feasible clustering with k_{max} clusters and a “coarsening” with k_{min} clusters there exists a feasible clustering **for every value** between k_{min} and k_{max}
- But you can’t always go from a clustering with k_{max} to one with k_{min} clusters if you perform closest cluster merge.
- That is if you use traditional agglomerative algorithms your dendrogram can end prematurely.

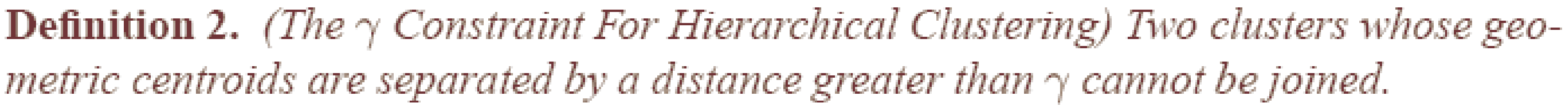
Dead-End Results

- For dead-end situations, you can’t use agglomerative clustering algorithms, otherwise you’ll prematurely terminate the dendrogram.

Constraint	**Dead-end Solutions?**
ML	No [PKDD05]
CL	Yes [PKDD05]
δ	No [PKDD05]
ε	No [PKDD05]

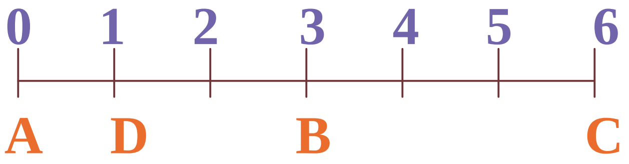
Constraint	**Dead-end Solutions?**
ML and ε	No [PKDD05]
ML and δ	No [PKDD05]
δ and ε	No [PKDD05]
ML, CL & ε	Yes [PKDD05]

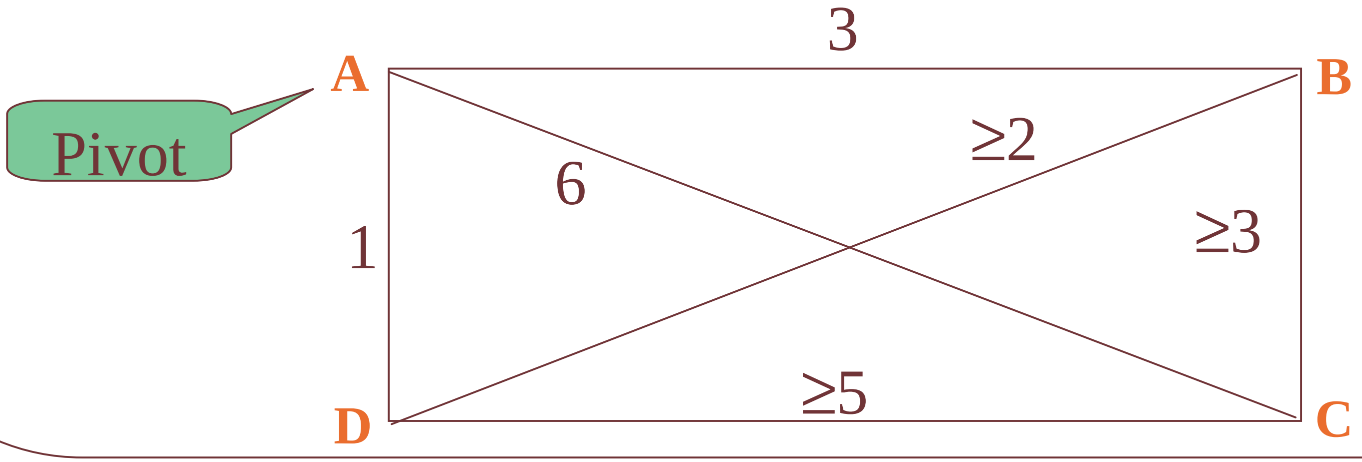
Speeding Up Agglomerative Clustering Using the Triangle Inequality - 1



Definition 2. *(The γ Constraint For Hierarchical Clustering) Two clusters whose geometric centroids are separated by a distance greater than γ cannot be joined.*

- Calculate distance between a pivot and all other points
- Bound distances on remaining pairs of points

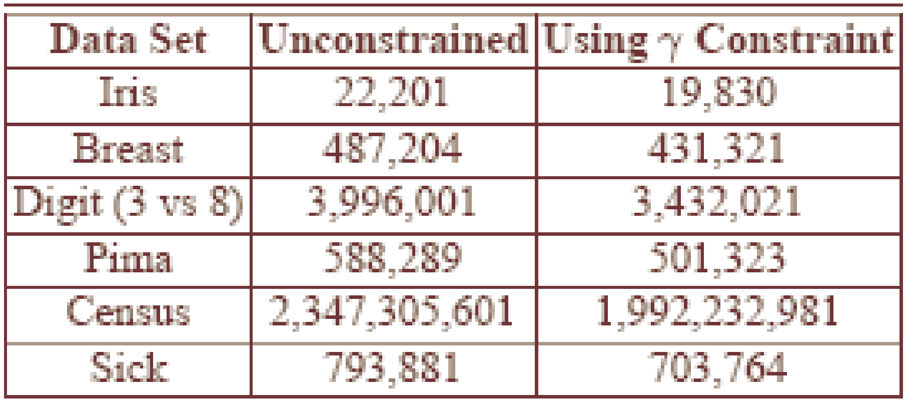




Speeding Up Agglomerative Clustering Using the Triangle Inequality - 2

Let $\gamma = 2$

$$D = \begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 3 & 6 & 1 \\ B & 3 & 0 & \xcancel{\geq 3} & \geq 2 \\ C & 6 & \xcancel{\geq 3} & 0 & \xcancel{\geq 5} \\ D & 1 & \geq 2 & \xcancel{\geq 5} & 0 \end{array}$$



Data Set	Unconstrained	Using γ Constraint
Iris	22,201	19,830
Breast	487,204	431,321
Digit (3 vs 8)	3,996,001	3,432,021
Pima	588,289	501,323
Census	2,347,305,601	1,992,232,981
Sick	793,881	703,764

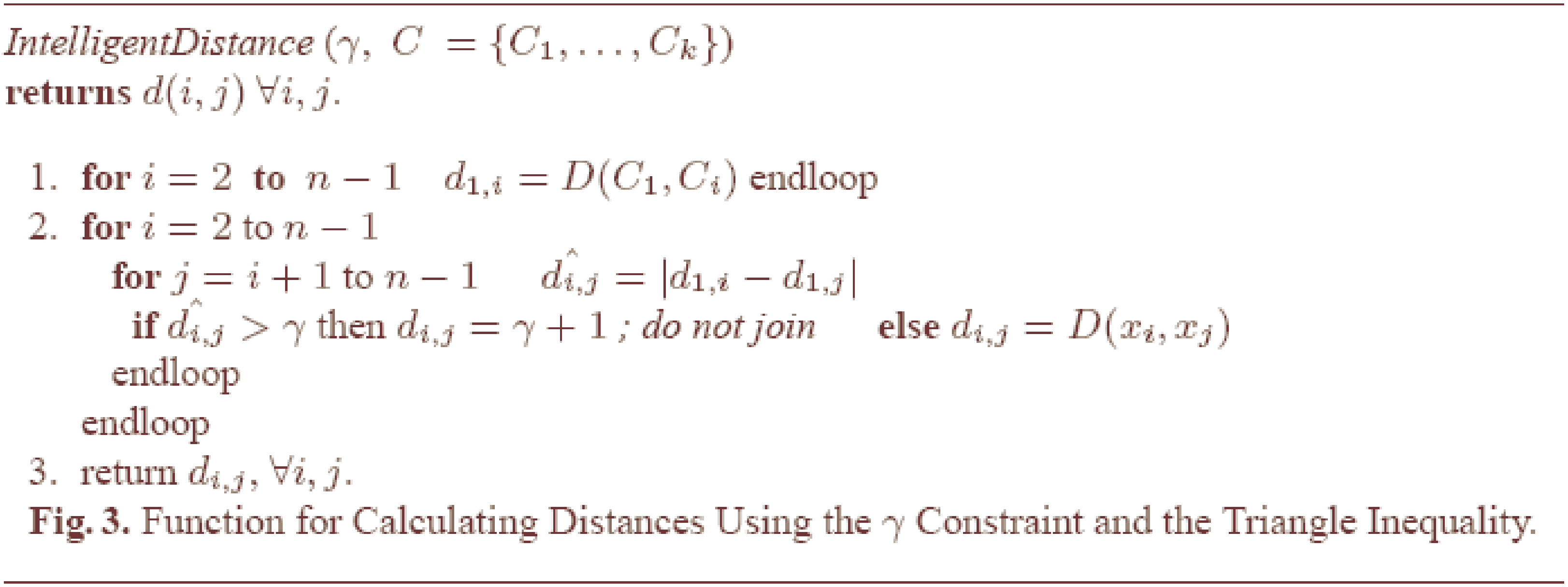
Mean number of distance calculations

Calculate: D(a,b)=1, D(a,c) = 3, D(a,d) = 6

Save D(b,d)≥5 D(c,d)≥3

Calculate D(b,c)≥2,

Algorithm



```
$IntelligentDistance\ (\gamma,\ C = \{C_1, \ldots, C_k\})$
returns $d(i,j)\ \forall i,j$.

 1. for $i = 2$ to $n-1$   $d_{1,i} = D(C_1, C_i)$ endloop
 2. for $i = 2$ to $n-1$
       for $j = i+1$ to $n-1$   $\hat{d_{i,j}} = |d_{1,i} - d_{1,j}|$
        if $\hat{d_{i,j}} > \gamma$ then $d_{i,j} = \gamma + 1$ ; do not join   else $d_{i,j} = D(x_i, x_j)$
       endloop
    endloop
 3. return $d_{i,j}, \forall i,j$.
```

Fig. 3. Function for Calculating Distances Using the γ Constraint and the Triangle Inequality.

- Worst case result $O(n^2)$ distance calculations
- Best case calculated bound **always** exceeds γ : $O(n)$
- Average case using the Markov inequality: save $1/2c$ distance calculations where $\gamma = c\rho$ and ρ is the average distance between two points.

$$P(X \geq A) \leq E[X] / A$$

Outline

- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
    - Enforcing constraints [Ian]
    - Hierarchical [Ian]
    - Learning distances [Sugato]
    - Initializing and pre-processing [Sugato]
    - Graph-based [Sugato]

Distance Learning as Convex Optimization [Xing et al. ’02]

- Learns a parameterized Mahalanobis (weighted Euclidean) distance using semi-definite programming (SDP):

$$\min_{A} \sum_{(s_i,s_j)\in ML} \| s_i - s_j \|_A^2 = \min_{A} \sum_{(s_i,s_j)\in ML} (s_i - s_j)^T A (s_i - s_j)$$

$$\sum_{(s_i,s_j)\in CL} \| s_i - s_j \|_A \geq 1$$

$$s.t. \qquad A \phi 0$$

$x^T = \{2,3\}, y^T = \{4,5\}: D_I(x,y) \quad \propto \{2-4, 3-5\}^T I \{2-4, 3-5\}$

$\propto \{2-4, 3-5\}^T \{I_{1,1}(2-4), I_{2,2}(3-5)\}$

$D_A(x,y) \quad \propto \{2-4, 3-5\}^T A \{2-4, 3-5\}$

$\propto A_{1,1}(2-4)^2 + A_{2,2}(3-5)^2$

Alternate formulation

- Equivalent optimization problem:

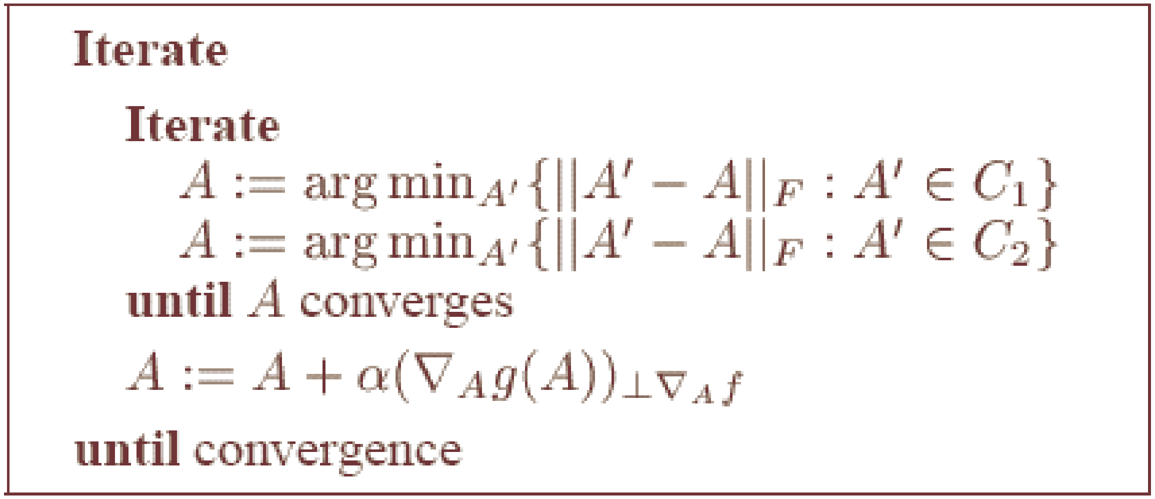
$$\max_{A} \quad g(A) = \sum_{(s_i, s_j) \in CL} \| s_i, s_j \|_A$$

$$f(A) = \sum_{(s_i, s_j) \in ML} \| s_i, s_j \|_A^2 \leq 1 \quad \longrightarrow C_1$$

$$s.t. \qquad A \phi 0 \qquad \longrightarrow C_2$$

Optimization Algorithm

- Solve optimization problem using combination of
  - gradient ascent: to optimize the objective
  - iterated projection algorithm: to satisfy the constraints

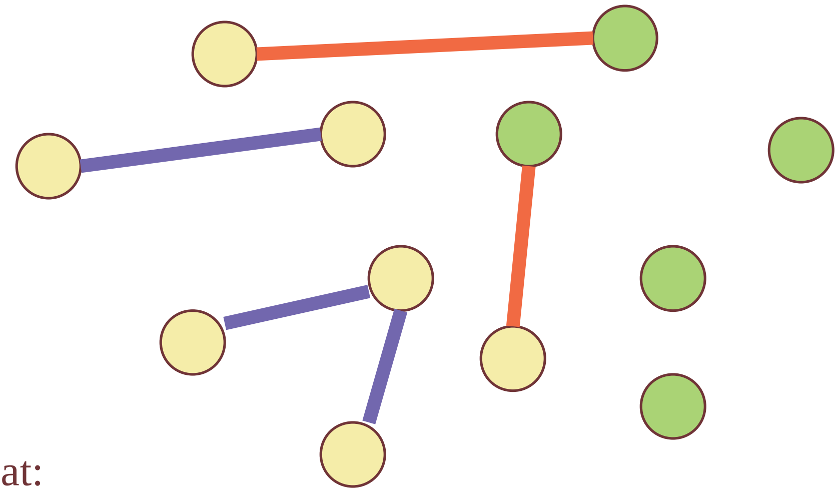


```
Iterate
    Iterate
        $A := \arg\min_{A'} \{||A' - A||_F : A' \in C_1\}$
        $A := \arg\min_{A'} \{||A' - A||_F : A' \in C_2\}$
    until $A$ converges
    $A := A + \alpha(\nabla_A g(A))_{\perp \nabla_A f}$
until convergence
```

- [Bie et al. ’05] use a variant of Linear Discriminant Analysis (LDA) to find semi-supervised metric more efficiently than SDP

Distance Learning in Product Space

[Hertz et al. ’04]



- Input:
  - Data set X in R^n.
  - Equivalence constraints
- Output: function D: $\underbrace{X \times X}_{\text{product space}} \rightarrow [0,1]$ such that:
  - points from the same class are close to each other.
  - points from different classes are very far from each other.
- Basic Observation:
  - *Equivalence constraints* ⇔ Binary labels in product space
  - Use boosting on product space to learn function

Boosting in a nutshell

A standard ML method that attempts to boost the performance of “weak” learners

Basic idea:

1. Initially, weights are set **equally**
2. **Iterate:**
   1. **Train** weak learner on weighted data
   2. **Increase** weights of **incorrectly** classified examples (force weak learner to focus on difficult examples)
3. Final hypothesis: **combination of weak hypotheses**

EM on Gaussian Mixture Model

- GMM: Standard data representation that models data using a number of Gaussian sources

- The parameters of the sources are estimated using the EM algorithm:

  – E step: Calculate Expected log-likelihood of the data over all possible assignments of data-points to sources

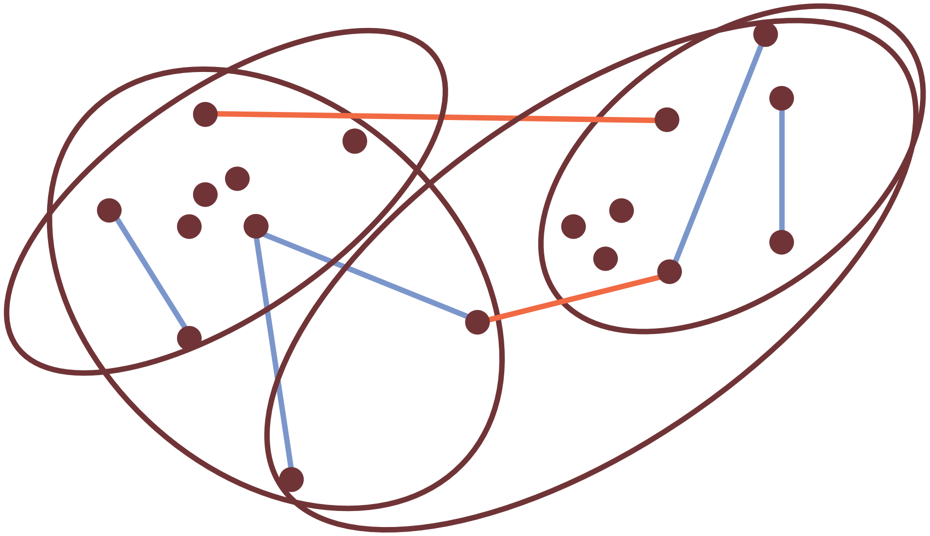
  – M step: Differentiate the Expectation w.r.t. the **parameters**

The Weak Learner: Constrained EM

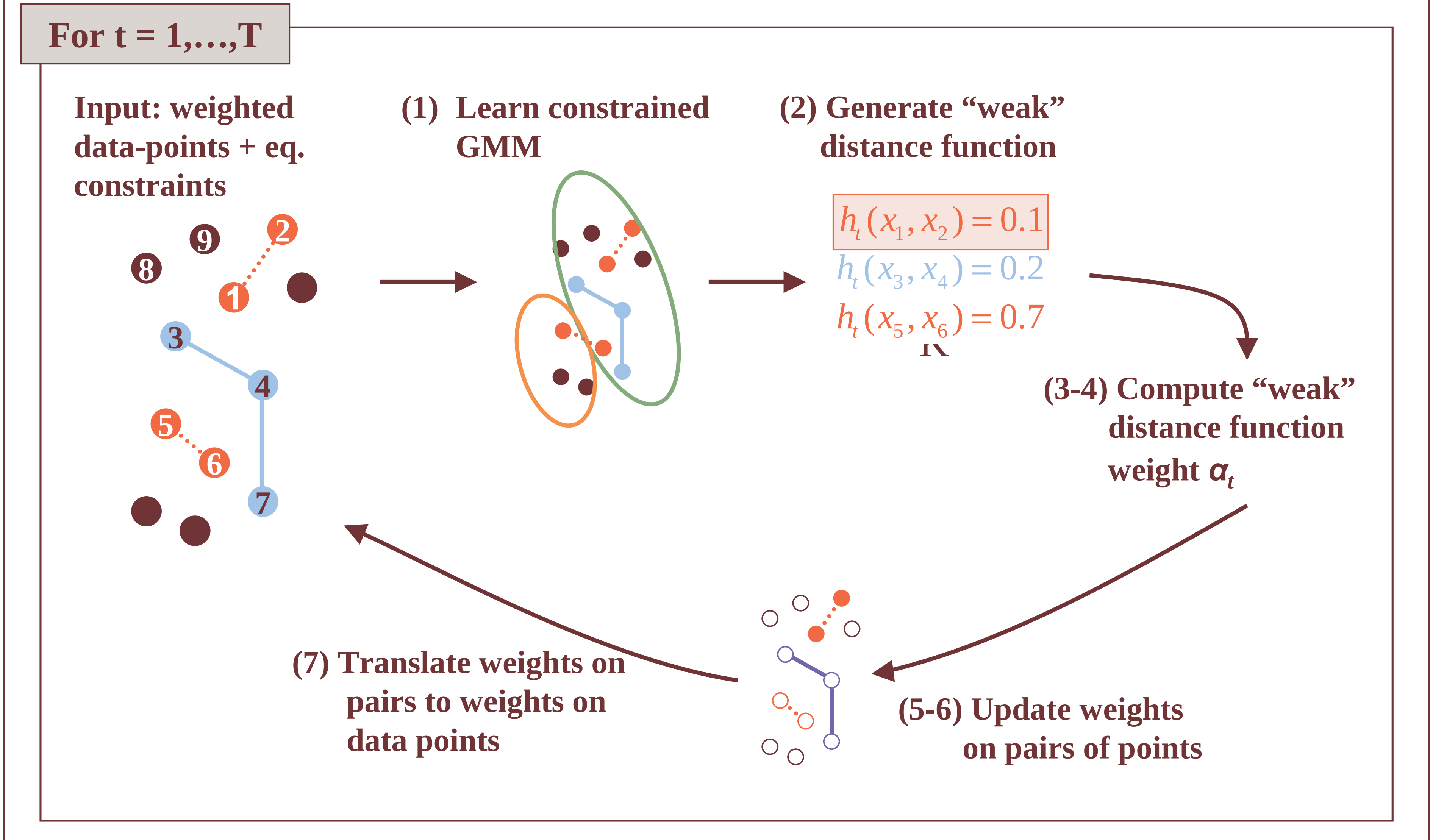
Constrained EM algorithm: fits a mixture of Gaussians to unlabeled data given a set of equivalence constraints.

Modification in case of equivalence constraints:

E step: sum only over assignments which comply with the constraints



The *DistBoost* algorithm



For t = 1,…,T

Input: weighted data-points + eq. constraints

(1) Learn constrained GMM

(2) Generate “weak” distance function

$h_t(x_1, x_2) = 0.1$

$h_t(x_3, x_4) = 0.2$

$h_t(x_5, x_6) = 0.7$

(3-4) Compute “weak” distance function weight α_t

(5-6) Update weights on pairs of points

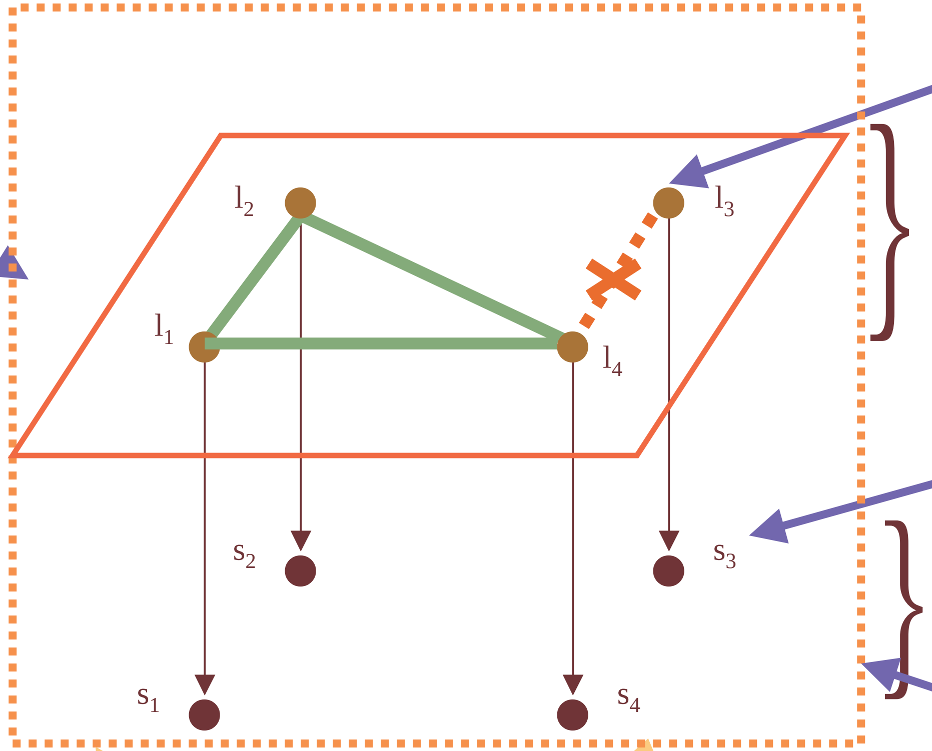
(7) Translate weights on pairs to weights on data points

Final distance function: $D(x_i, x_j) = \sum_{t=1}^{T} \alpha_t h_t(x_i, x_j)$

Integrated Approach: HMRF

[Basu et al. '04]

Markov Random Field (MRF)



Hidden RVs of cluster labels: *L*

P(L): Prior over constraints

Observed data values: *S*

P(S|L): Data Likelihood

Hidden Markov Random Field (HMRF)

Joint probability
P(L,S) = P(L).P(S|L)

Goal of constrained clustering: estimation of P(L,S) on HMRF

Cannot-link

Must-link

Constrained Clustering on HMRF

Gibbs potential for constraints

$$\Pr(L) \propto \exp[-\sum_{i,j} V(s_i, s_j, l_i, l_j)]$$

Cluster distortion

$$\Pr(S \mid L) \propto \exp[-\sum_{s_i} D(s_i, C_{l_i})]$$

Joint probability

$$\Downarrow$$

$$\Pr(L , S) = \Pr(S \mid L) \Pr(L)$$

Overall objective of constrained clustering

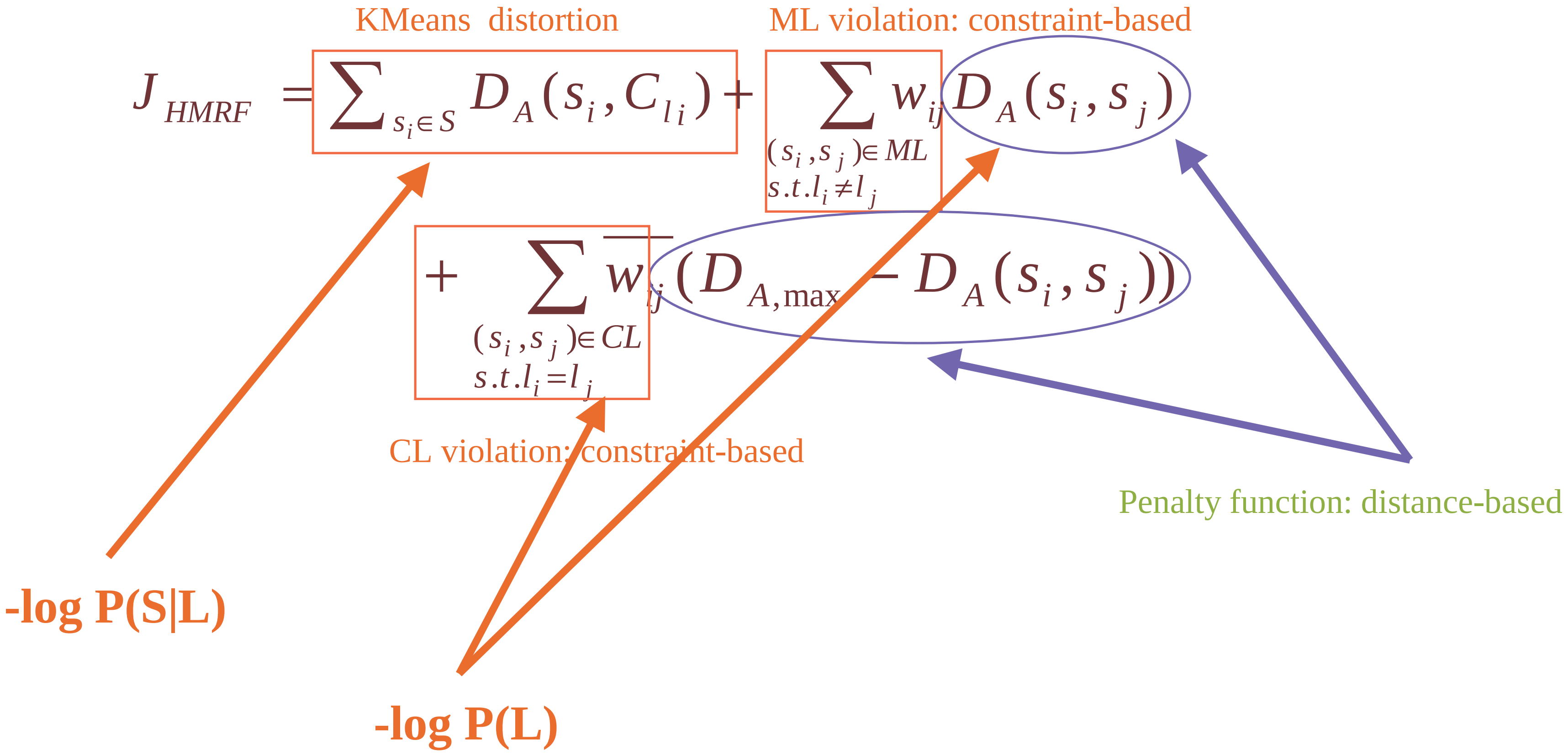
$$-\log \Pr(L, S) \propto \left(\sum_{s_i} D(s_i, C_{l_i}) + \sum_{i,j} V(s_i, s_j, l_i, l_j) \right)$$

MRF potential

- Generalized Potts (Ising) potential:

$$V(s_i, s_j, l_i, l_j) = \left\{ \begin{array}{ll} w_{ij} D_A(s_i, s_j) & if \quad l_i \neq l_j, (s_i, s_j) \in ML \\ \overline{w_{ij}} \left[D_{A,\max} - D_A(s_i, s_j) \right] & if \quad l_i = l_j, (s_i, s_j) \in CL \\ 0 & else \end{array} \right\}$$

HMRF-KMeans: Objective Function



KMeans distortion

ML violation: constraint-based

$$J_{HMRF} = \boxed{\sum_{s_i \in S} D_A(s_i, C_{l_i})} + \boxed{\sum_{\substack{(s_i, s_j) \in ML \\ s.t. l_i \neq l_j}} w_{ij}} D_A(s_i, s_j)$$

$$+ \boxed{\sum_{\substack{(s_i, s_j) \in CL \\ s.t. l_i = l_j}} \overline{w}_{ij}} (D_{A,\max} - D_A(s_i, s_j))$$

CL violation: constraint-based

Penalty function: distance-based

-log P(S|L)

-log P(L)

HMRF-KMeans: Algorithm

Initialization:

- Use neighborhoods derived from constraints to initialize clusters

Till *convergence*:

1. **Point assignment:**

- Assign each point s to cluster h^* to minimize both distance and constraint violations (Note: this is greedy, other methods possible)

2. **Mean re-estimation:**

- Estimate cluster centroids C as means of each cluster
- Re-estimate parameters A of D_A to minimize constraint violations

HMRF-KMeans: Convergence

Theorem:

HMRF-KMeans converges to a local minima of J_{HMRF} for for Bregman divergences D (e.g., KL divergence, squared Euclidean distance) or directional distances (e.g., Pearson’s distance, cosine distance)

Ablation/Sensitivity Experiment

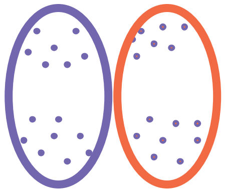
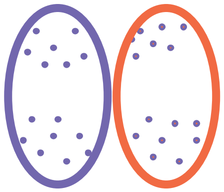
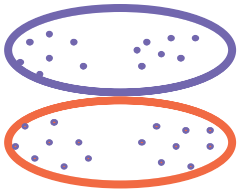
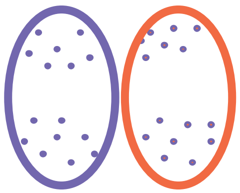
• MPCK-Means:	both constraints and distance learning
• MK-Means:	only distance learning
• PCK-Means:	only constraints
• K-Means:	purely unsupervised

Evaluation Measure

- Compare cluster partitioning to class labels on the dataset
- Mutual Information measure calculated only on test set

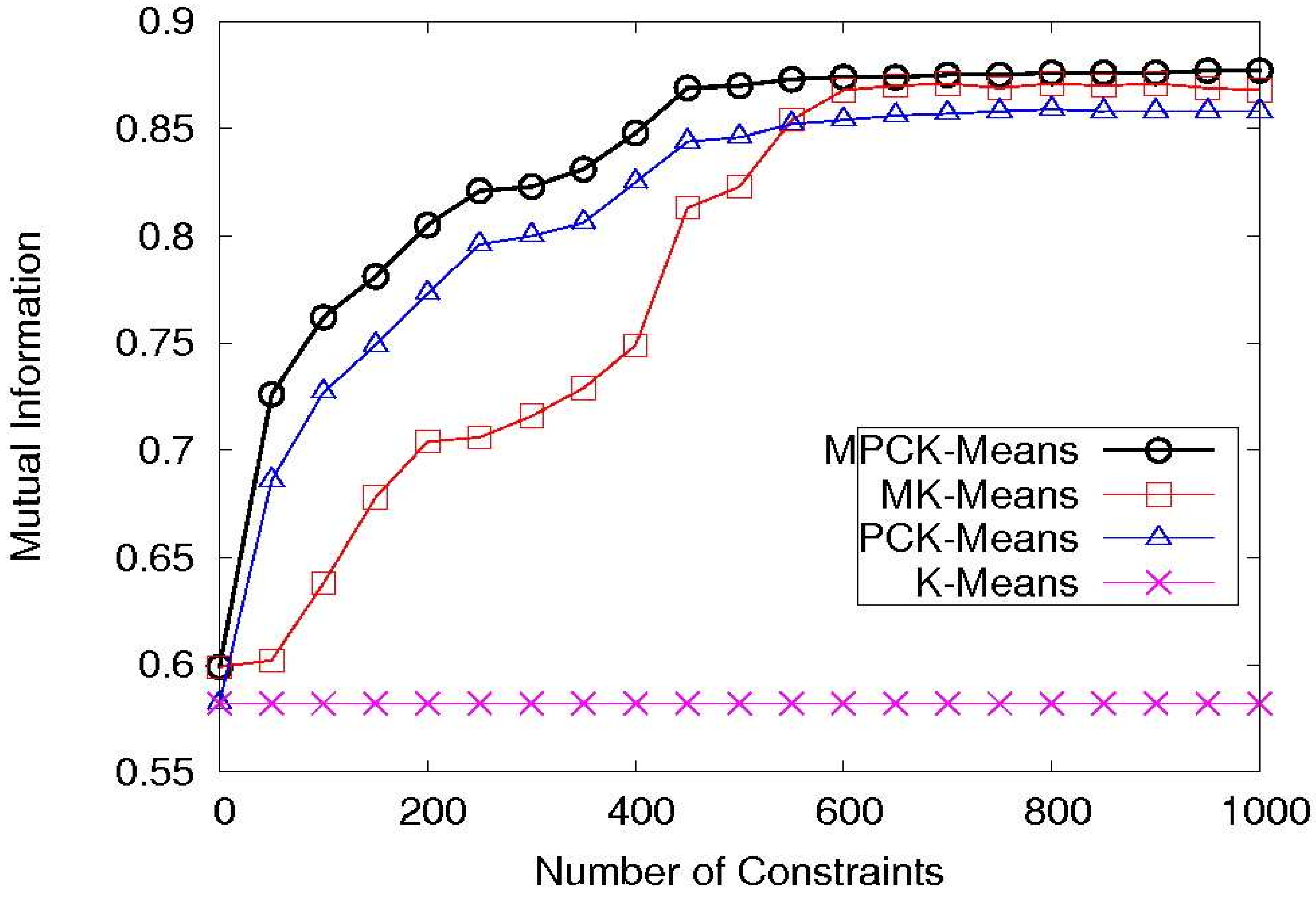
[Strehl et al. ’00]

$$MI = \frac{I(C;K)}{[H(C)+H(K)]/2}$$

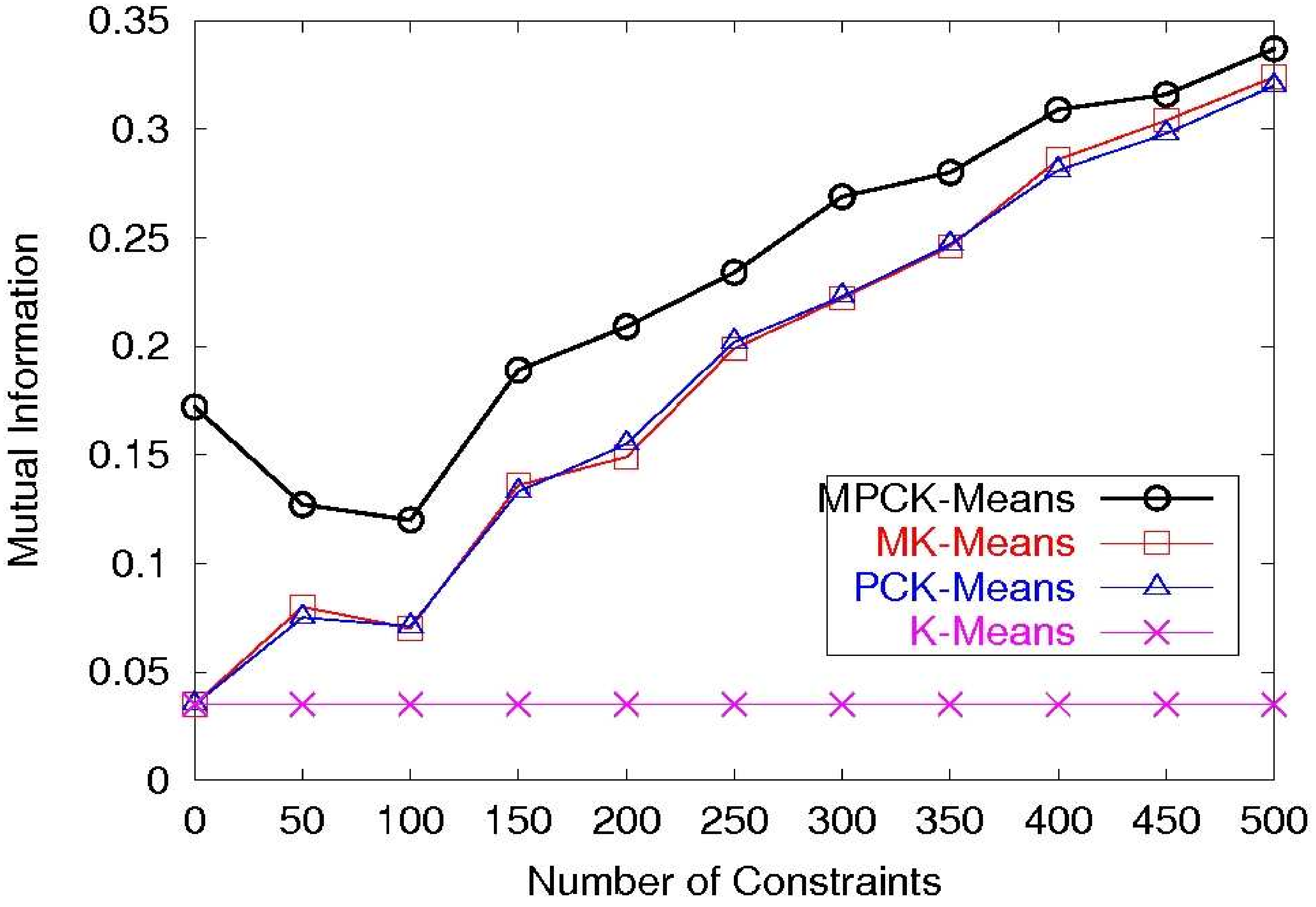
Cluster partitions	Underlying classes	MI value
		High
		Low

Experiment Results: PenDigits subset

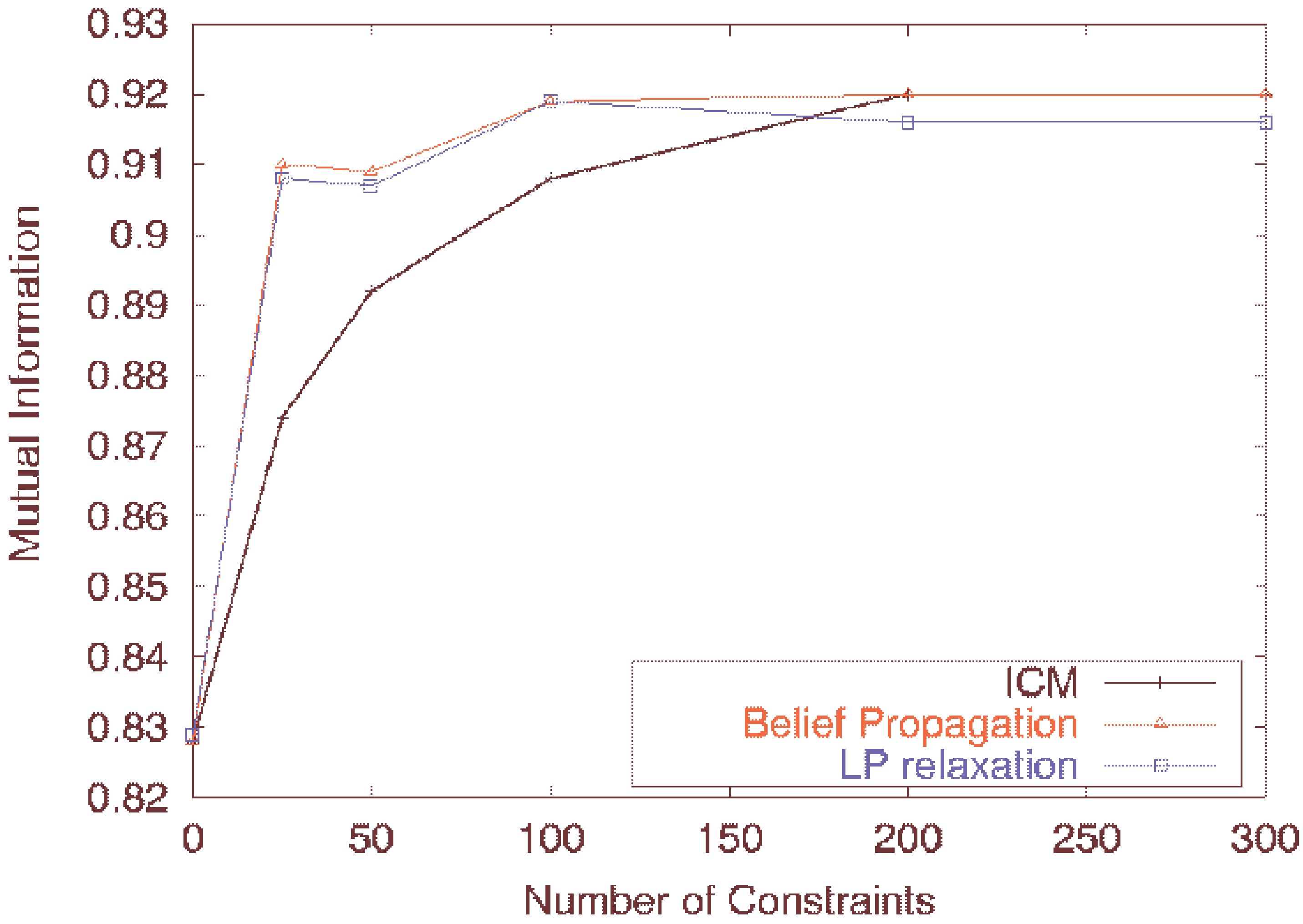
(squared Euclidean distance)



Experiment Results: 20Newsgroups-subset
(cosine distance)



Comparing Inference Techniques for HMRF



Related Formulations

- Maximum entropy EM
  - Incorporates prior knowledge in both labels and constraints
  - Modify the likelihood function:

$$\min_{\Theta}(\alpha L(X^u;\Theta)+\beta L(X^l;Y;\Theta)+(1-\alpha-\beta)L(X^c;C;\Theta))$$

  - Infer Gibbs potential from MaxEnt solution of *P(Y)* under constraints encoded in *L* and *C*
  - Generalizes K-Means formulation to EM
  - Replaces ICM for posterior distribution calculation in E-step by:
    - Mean-field approximation [Lange et al. ’05]
    - Gibbs sampling [Lu et al. ’05]

Outline

- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Finding Informative Constraints given a quota of Queries

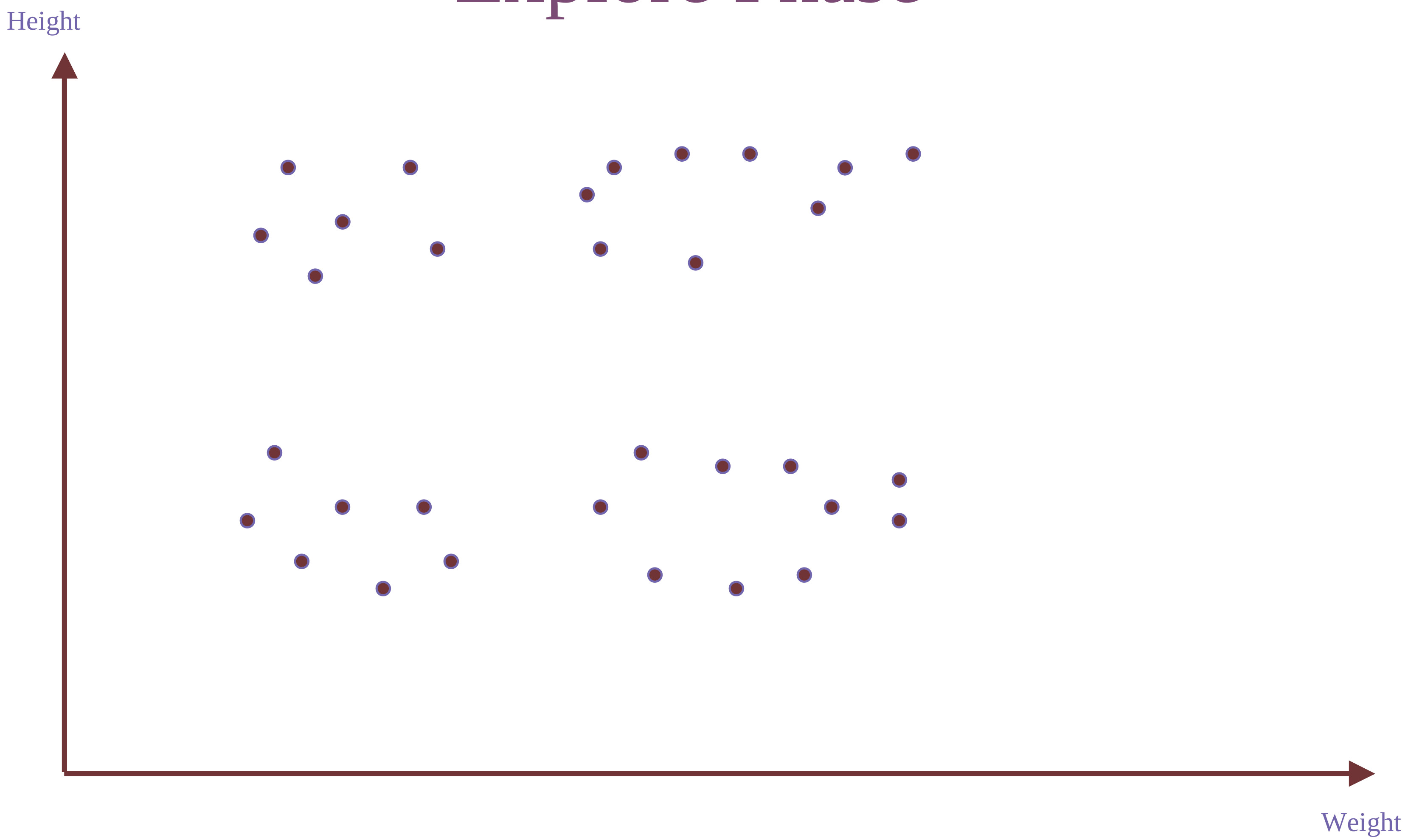
- Active learning for constraint acquisition [Basu et al.'04]:
  - In interactive setting, constraints obtained by queries to a user
  - Need to get **informative** constraints to get better clustering
- Two-phase active learning algorithm:
  - Explore: Use *farthest-first* traversal [Hochbaum et al.'85] to explore the data and find *K* pairwise-disjoint neighborhoods (cluster skeleton) rapidly
  - Consolidate: Consolidate basic cluster skeleton by getting more points from each cluster, within max *(K-1)* queries for any point
- Related technique [Cohn et al.'03] :
  - Can incorporate any user feedback to "repair" clustering metric

Algorithm: Explore

- Pick a point s at random, add it to neighborhood N_1 , $\lambda = 1$
- While queries are allowed and ($\lambda < k$)
  - Pick point s farthest from existing λ neighborhoods
  - If by querying s is *cannot-linked* to all existing neighborhoods, then set $\lambda = \lambda+1$, start new neighborhood N_λ with s
  - Else, add s to neighborhood with which it is *must-linked*

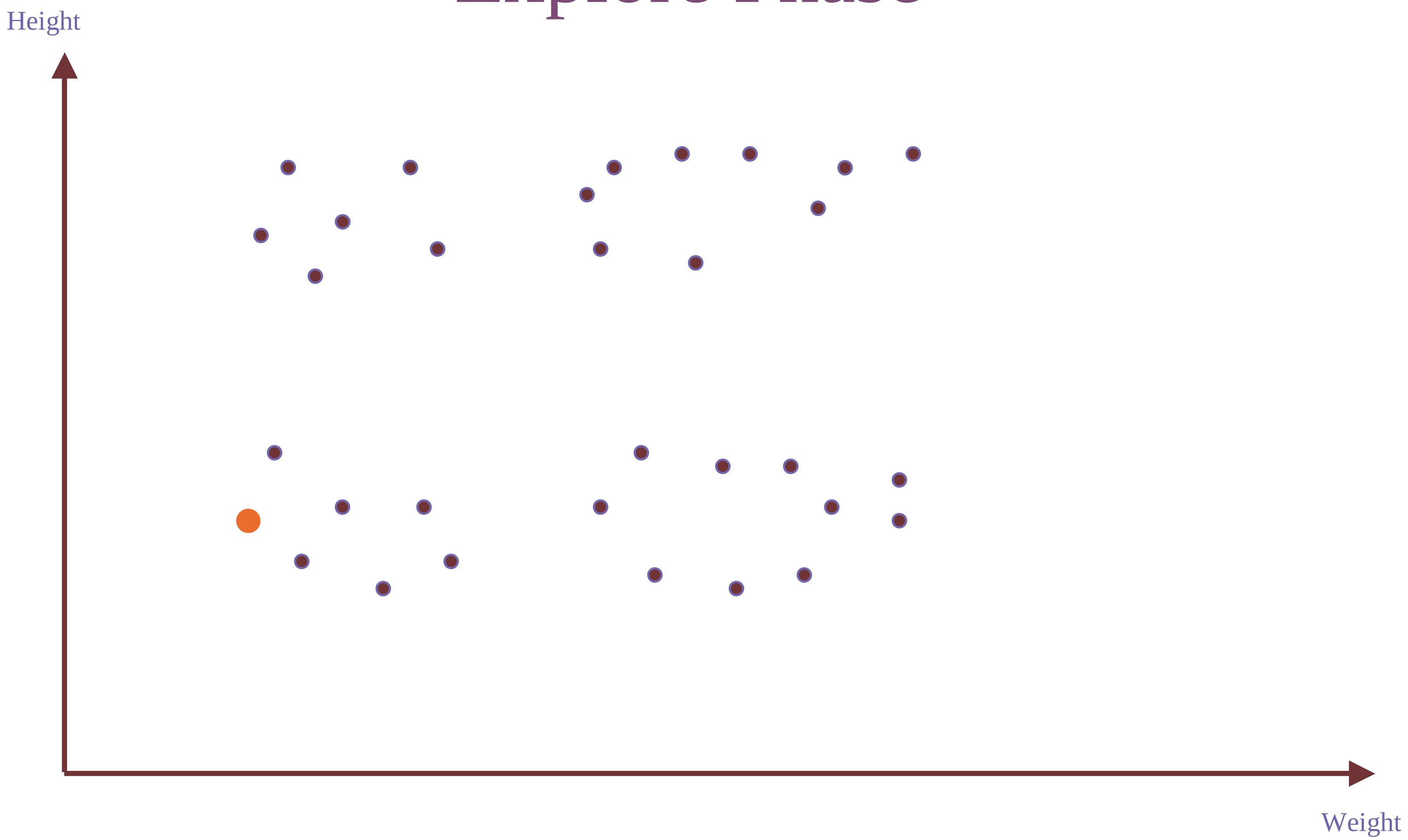
Active Constraint Acquisition for Clustering

Explore Phase



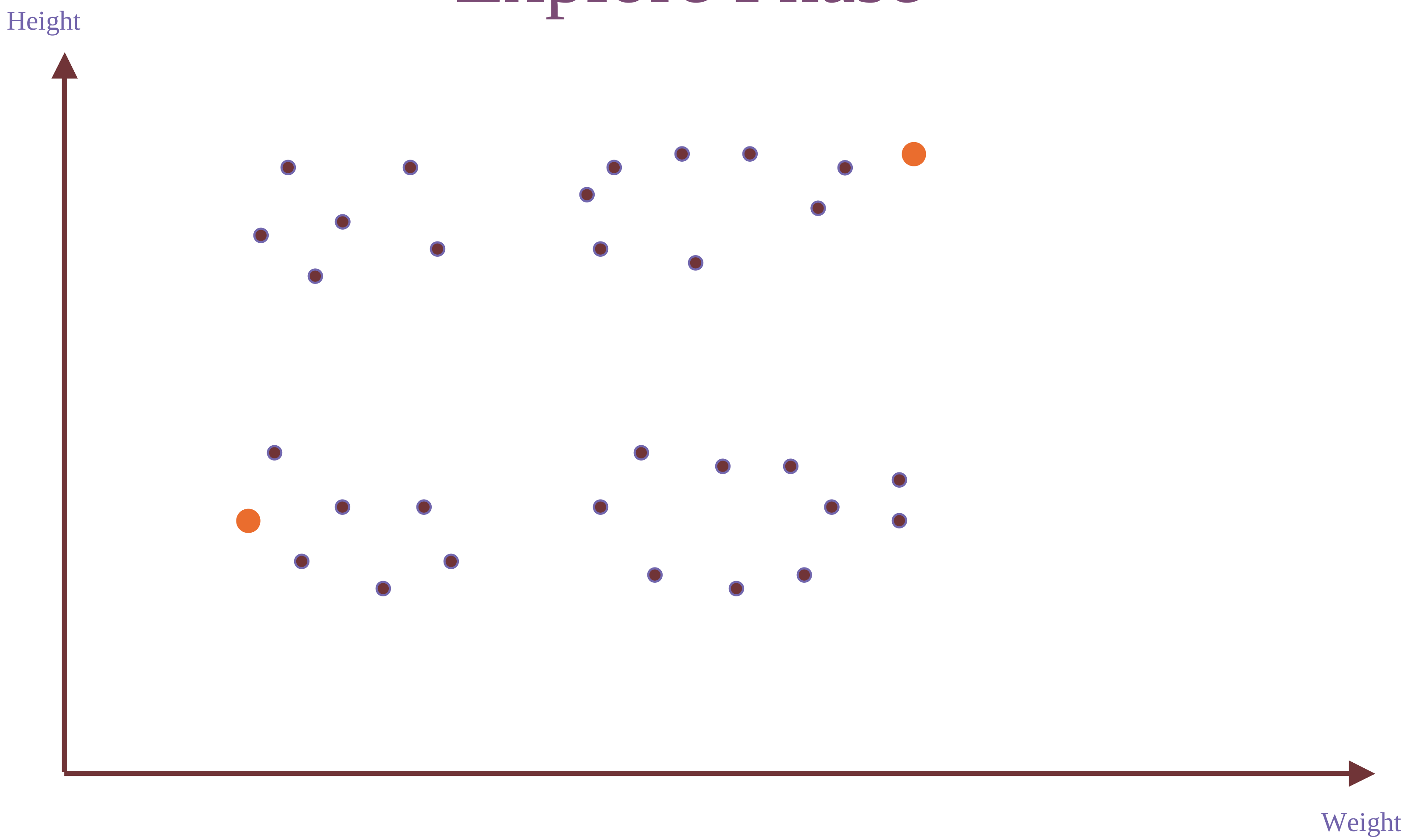
Active Constraint Acquisition for Clustering

Explore Phase



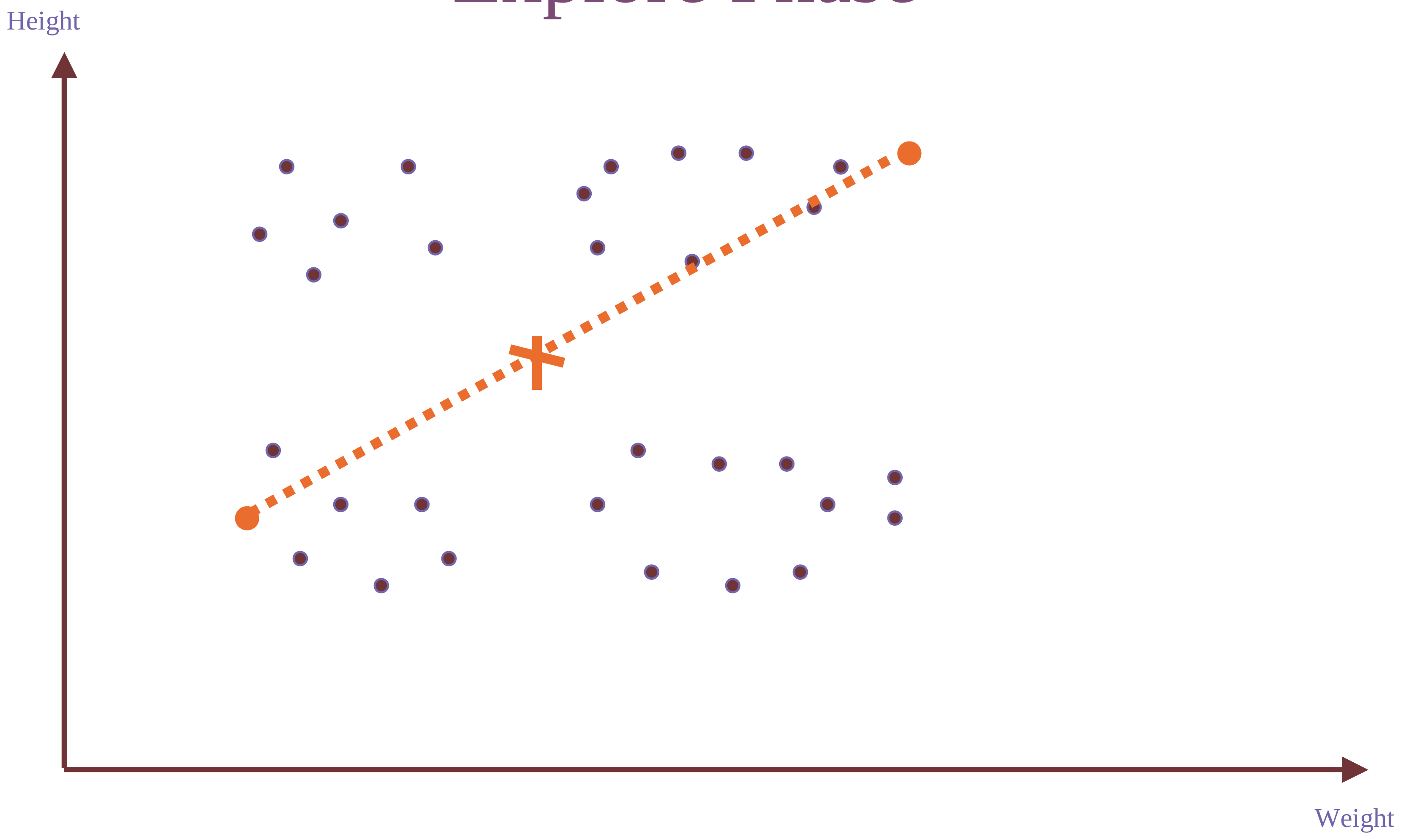
Active Constraint Acquisition for Clustering

Explore Phase



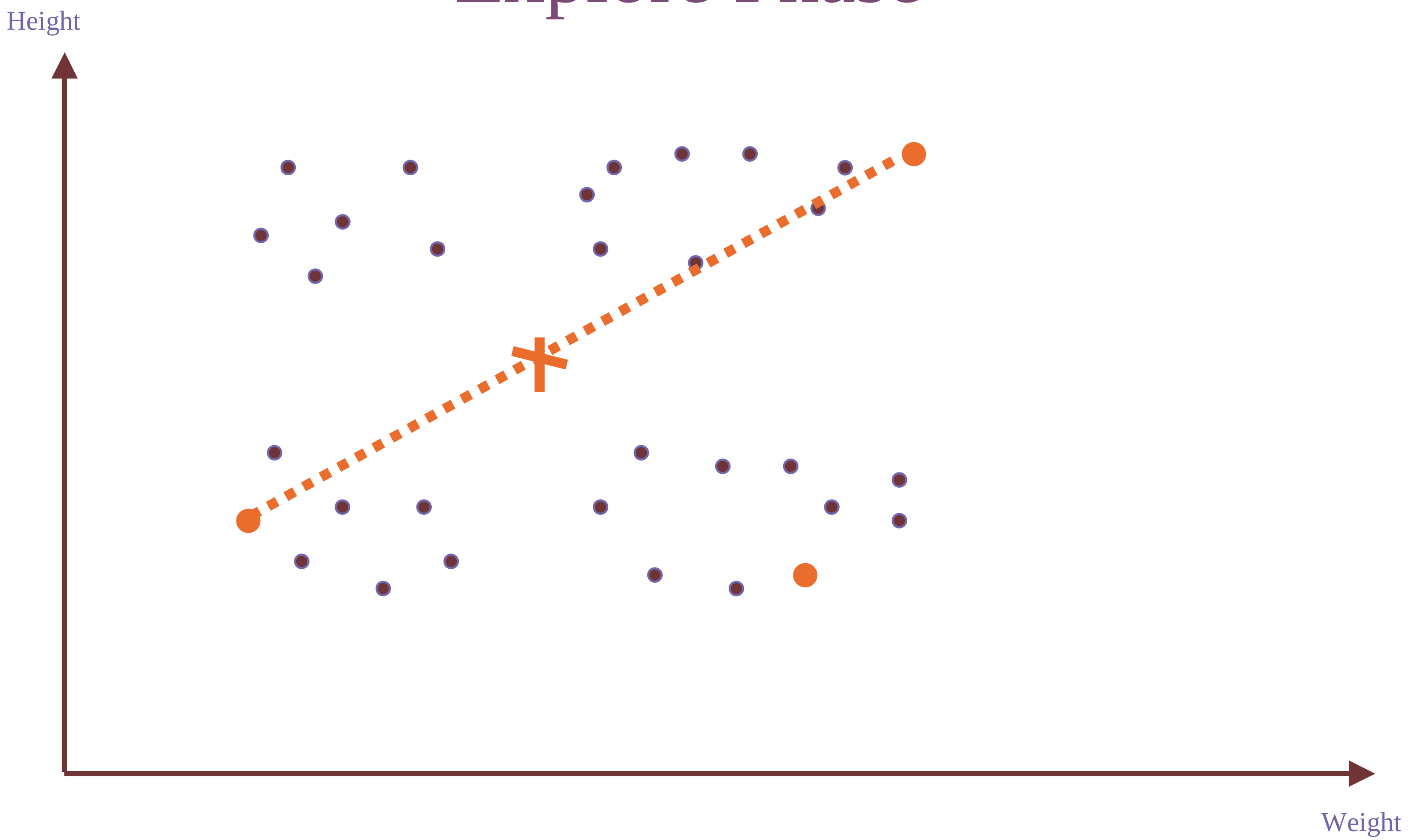
Active Constraint Acquisition for Clustering

Explore Phase



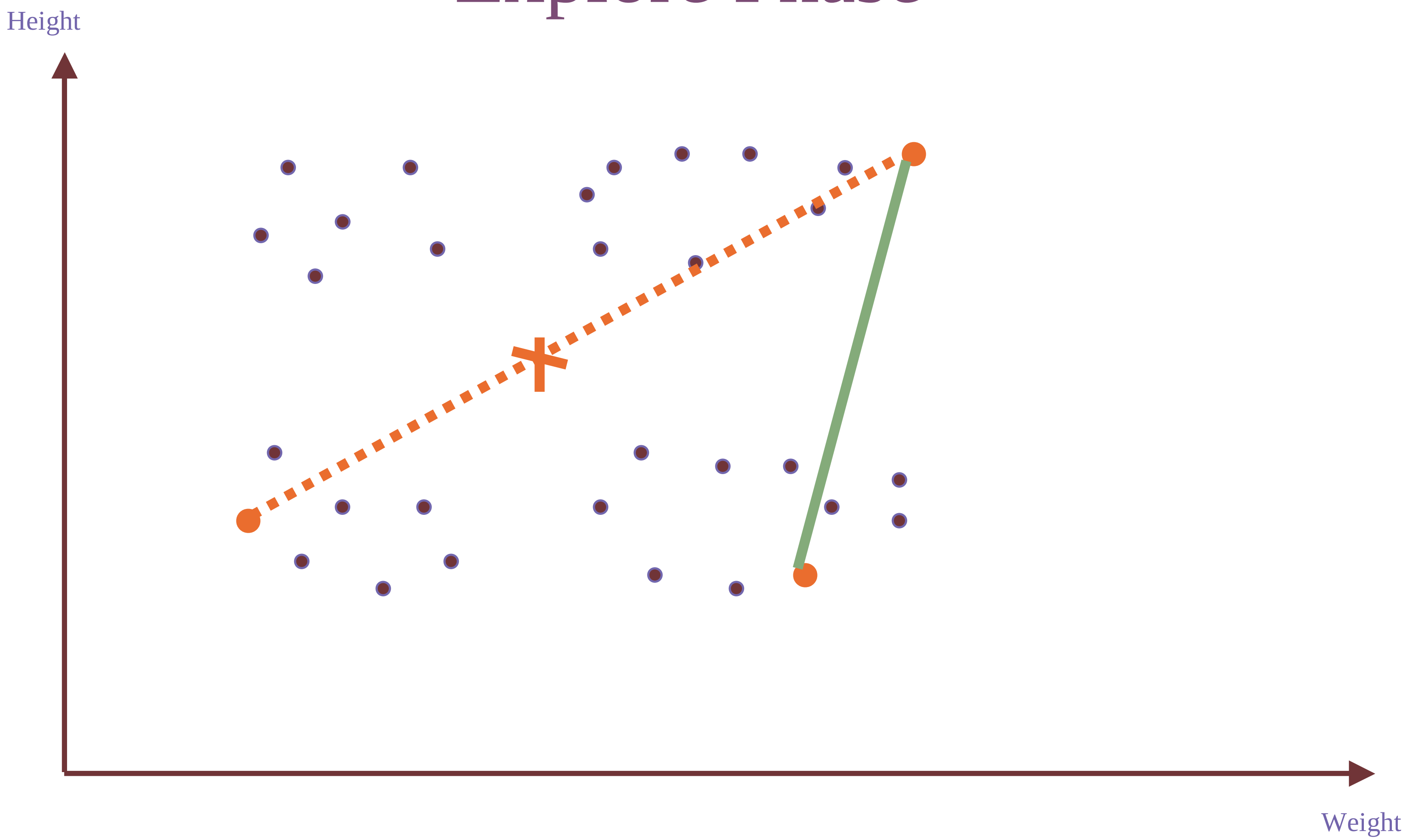
Active Constraint Acquisition for Clustering

Explore Phase



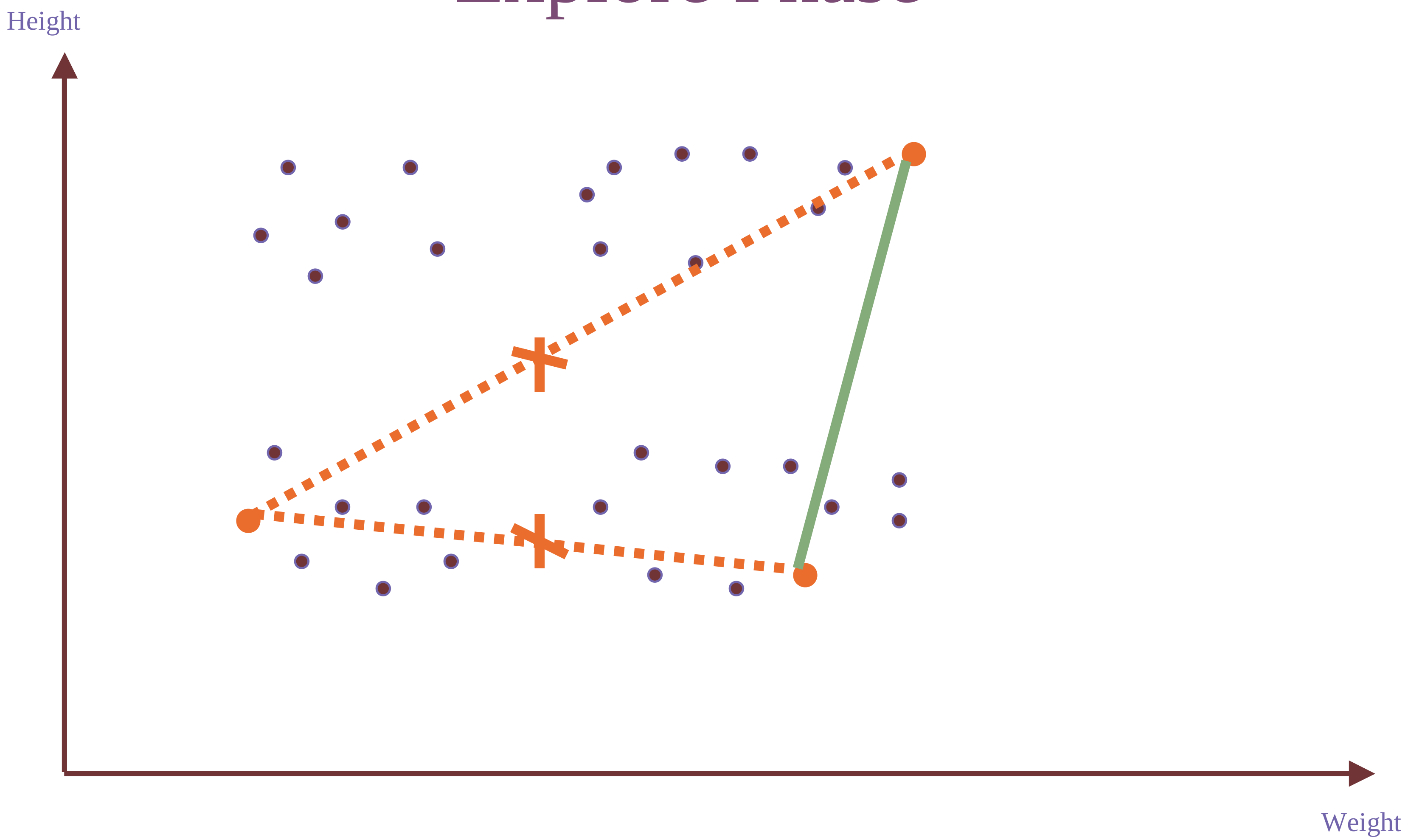
Active Constraint Acquisition for Clustering

Explore Phase



Active Constraint Acquisition for Clustering

Explore Phase

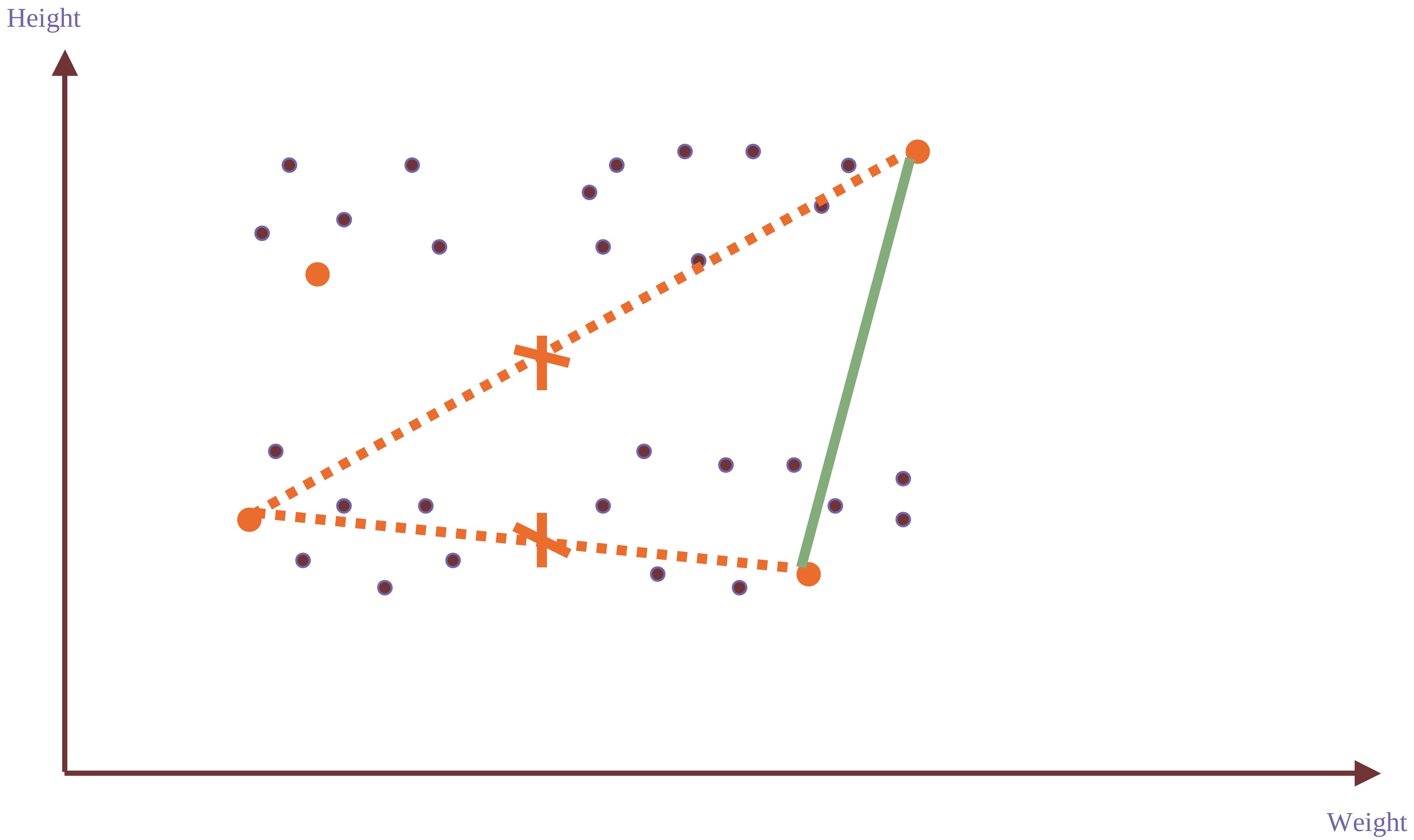


Algorithm: Consolidate

- Estimate centroids of each of the λ neighborhoods
- While queries are allowed
  - Randomly pick a point *s* not in the existing neighborhoods
  - Query *s* with each neighborhood (in sorted order of decreasing distance from *s* to centroids) until *must-link* is found
  - Add *s* to that neighborhood to which it is *must-linked*

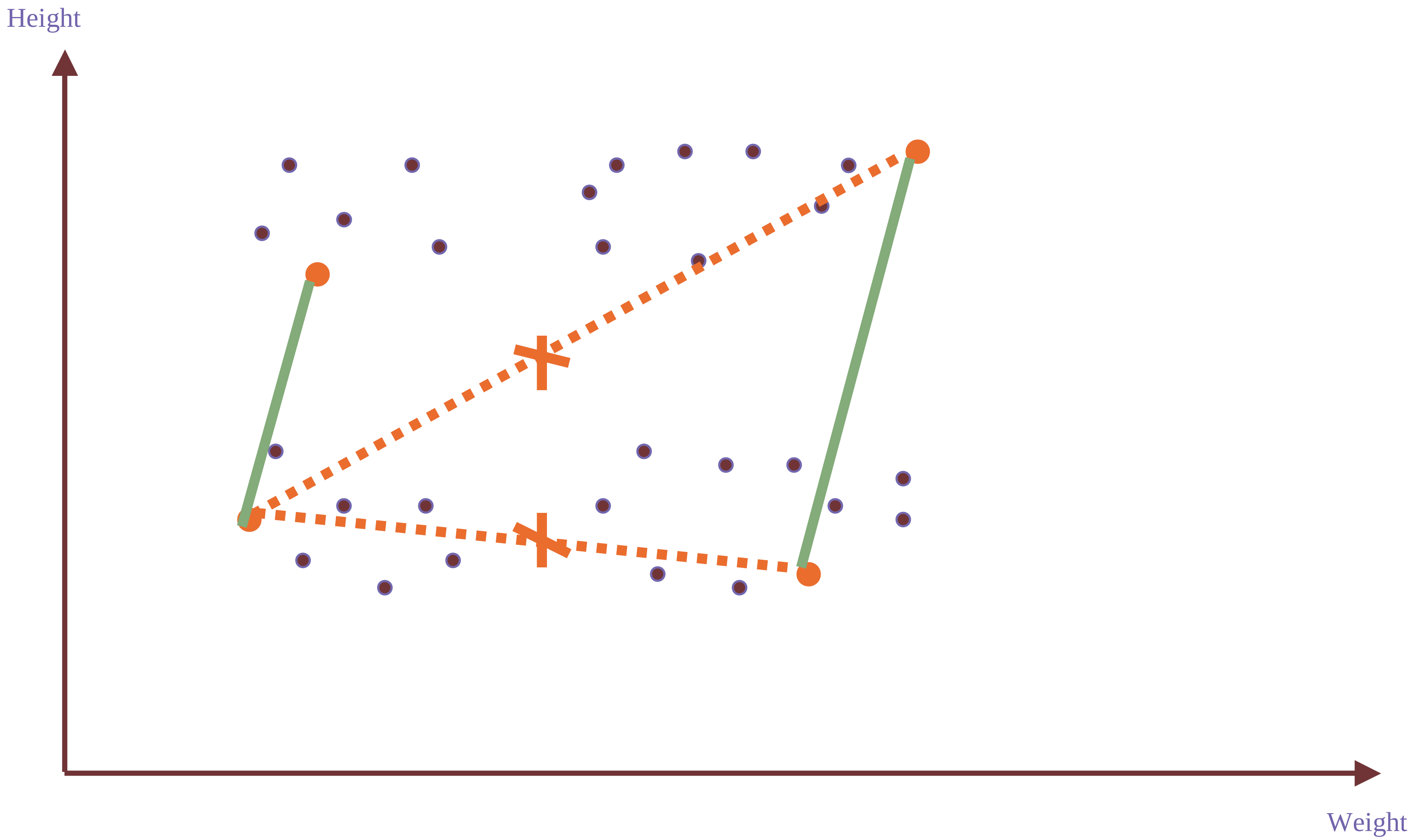
Active Constraint Acquisition for Clustering

Consolidate Phase



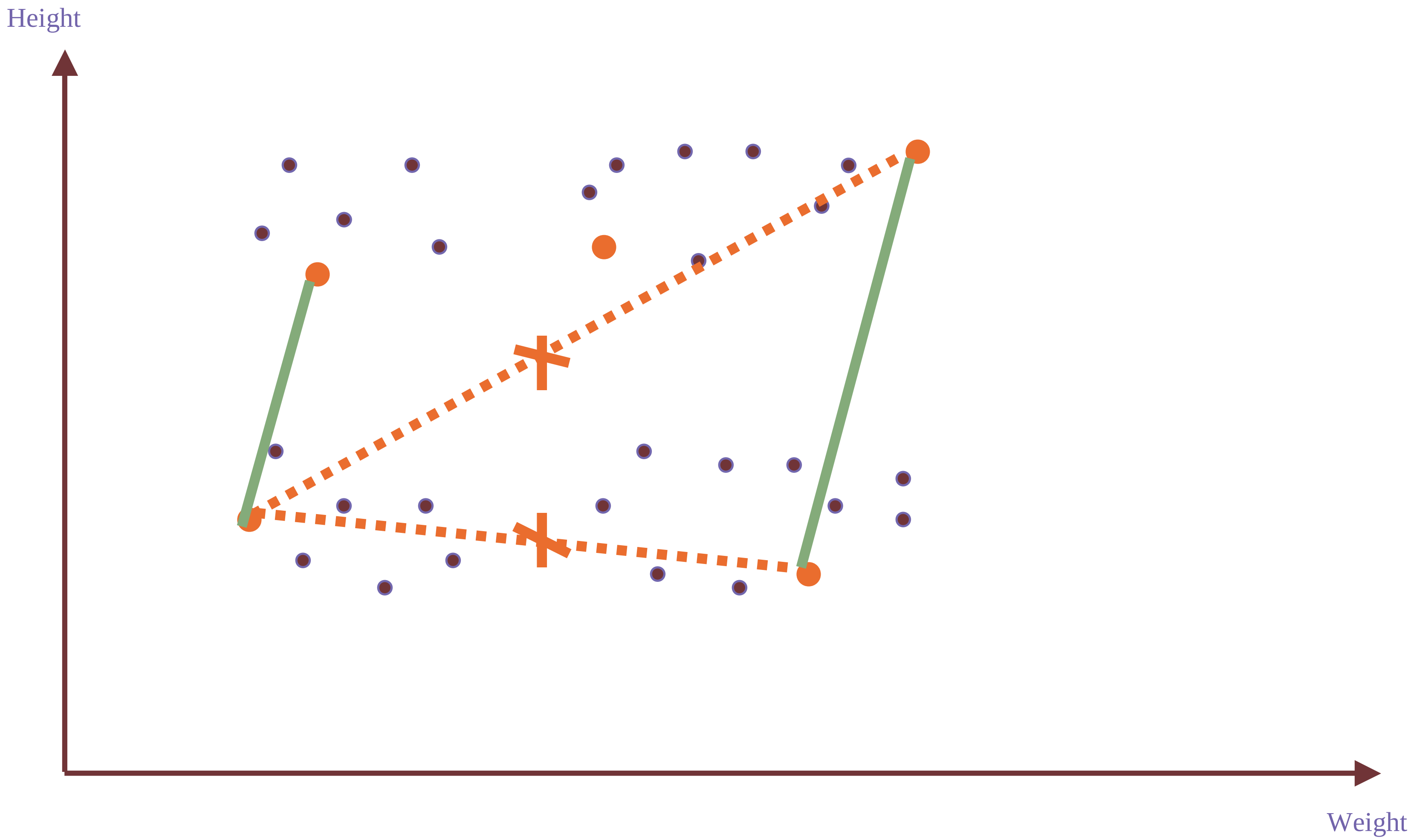
Active Constraint Acquisition for Clustering

Consolidate Phase



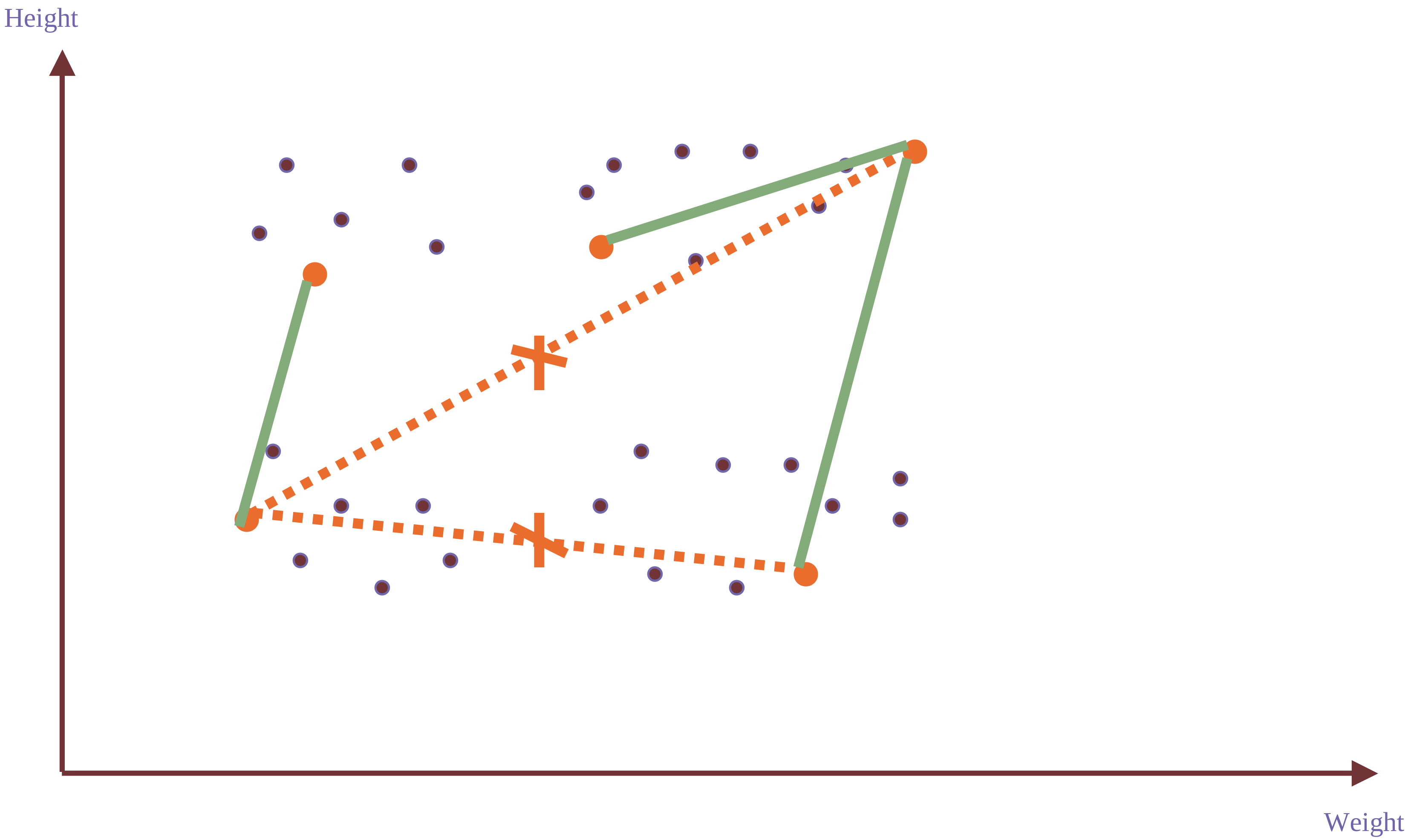
Active Constraint Acquisition for Clustering

Consolidate Phase



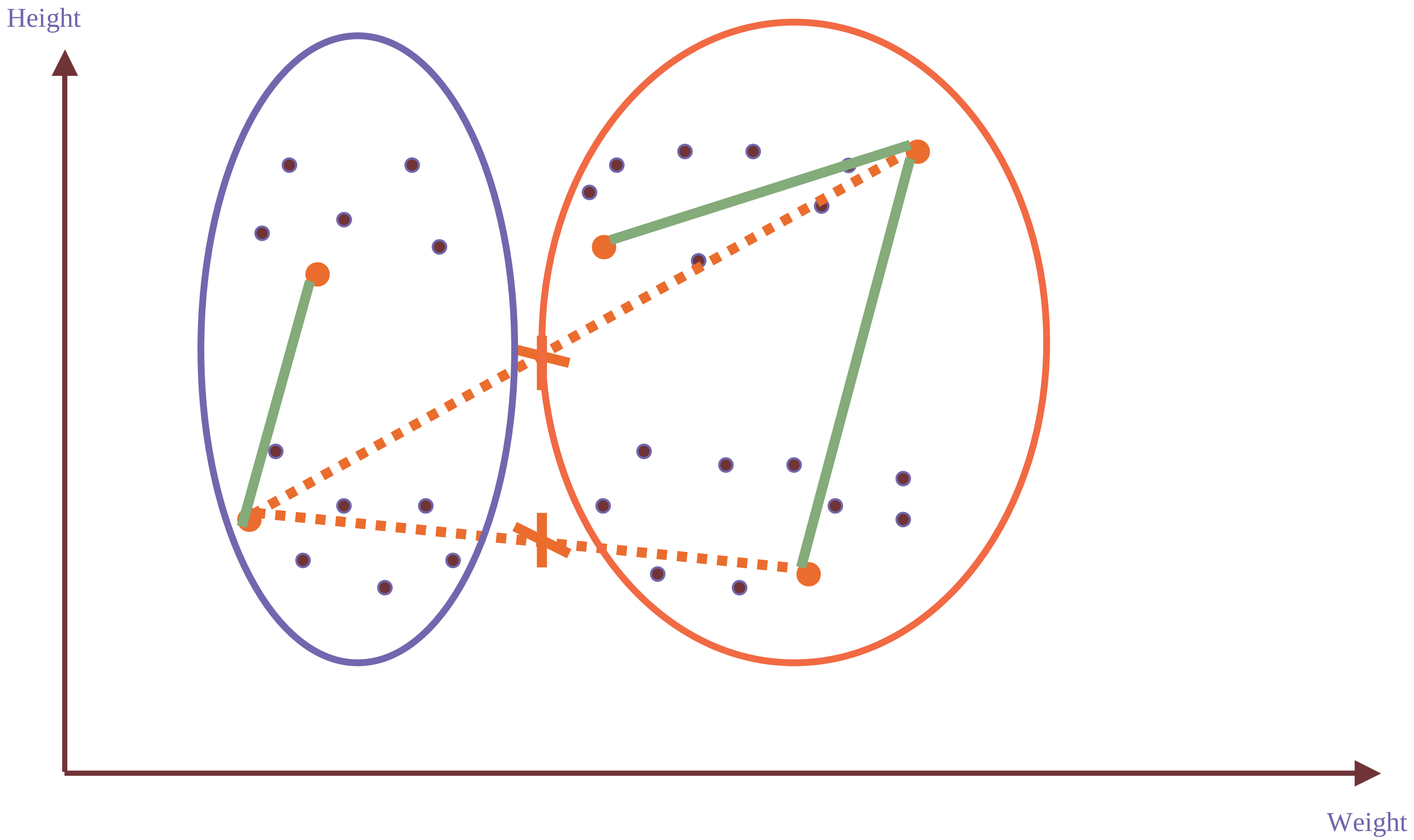
Active Constraint Acquisition for Clustering

Consolidate Phase

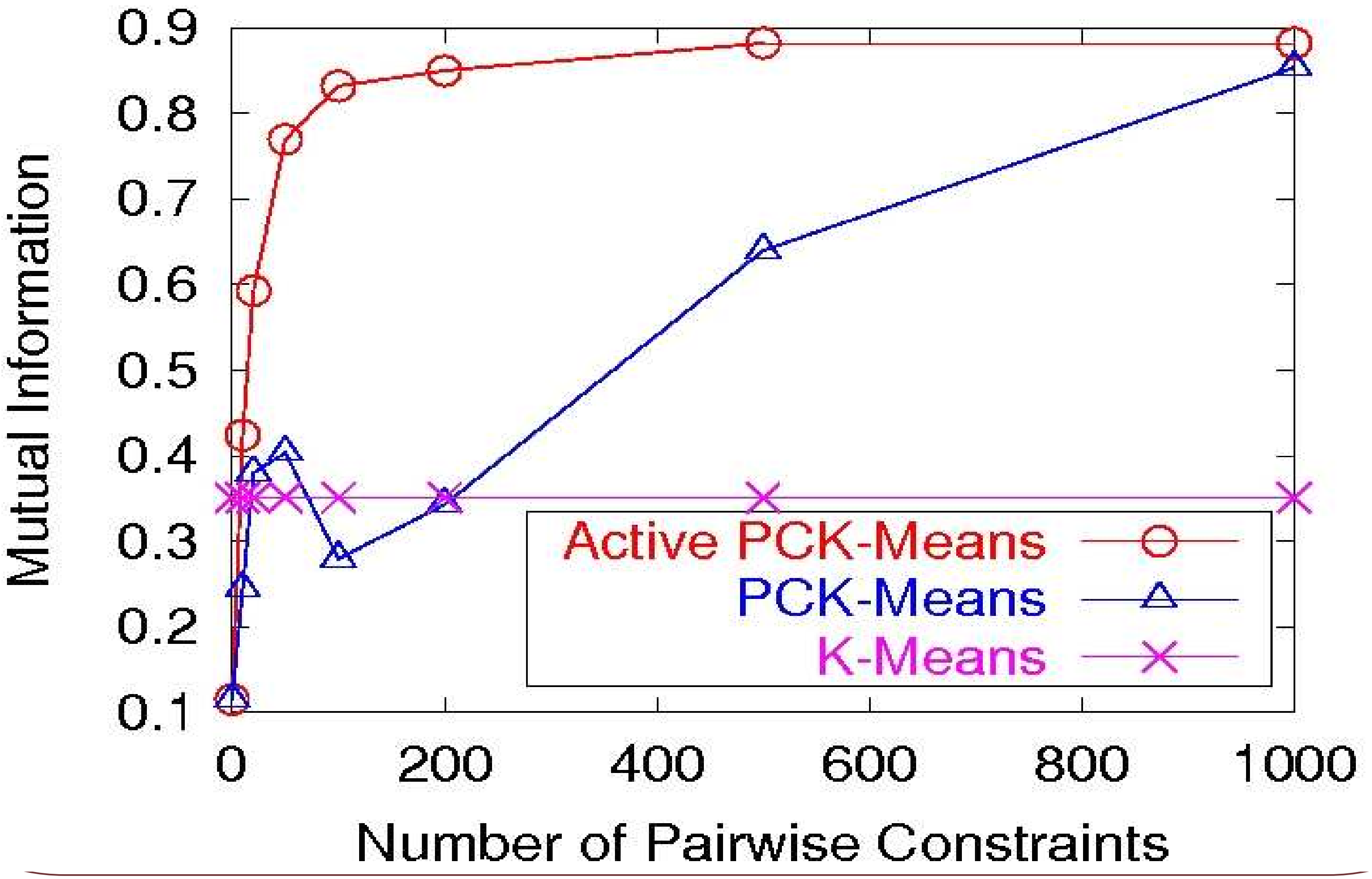


Active Constraint Acquisition for Clustering

Consolidate Phase



Experiments: 20-Newsgroups subset



Confusion Matrices

No constraints

	Cluster1	Cluster2	Cluster3
Misc	**71**	12	17
Guns	25	**61**	14
Mideast	12	36	**52**

20 queries

	Cluster1	Cluster2	Cluster3
Misc	**84**	7	9
Guns	5	**91**	4
Mideast	7	7	**86**

Algorithms to Seed K-Means When Feasibility Problem is in P [Davidson et al. ’05]

- Each algorithm will find a feasible solution.
- You can build upon each to make them minimize the vector quantization error (or what-ever objective function your algorithm has) as well.

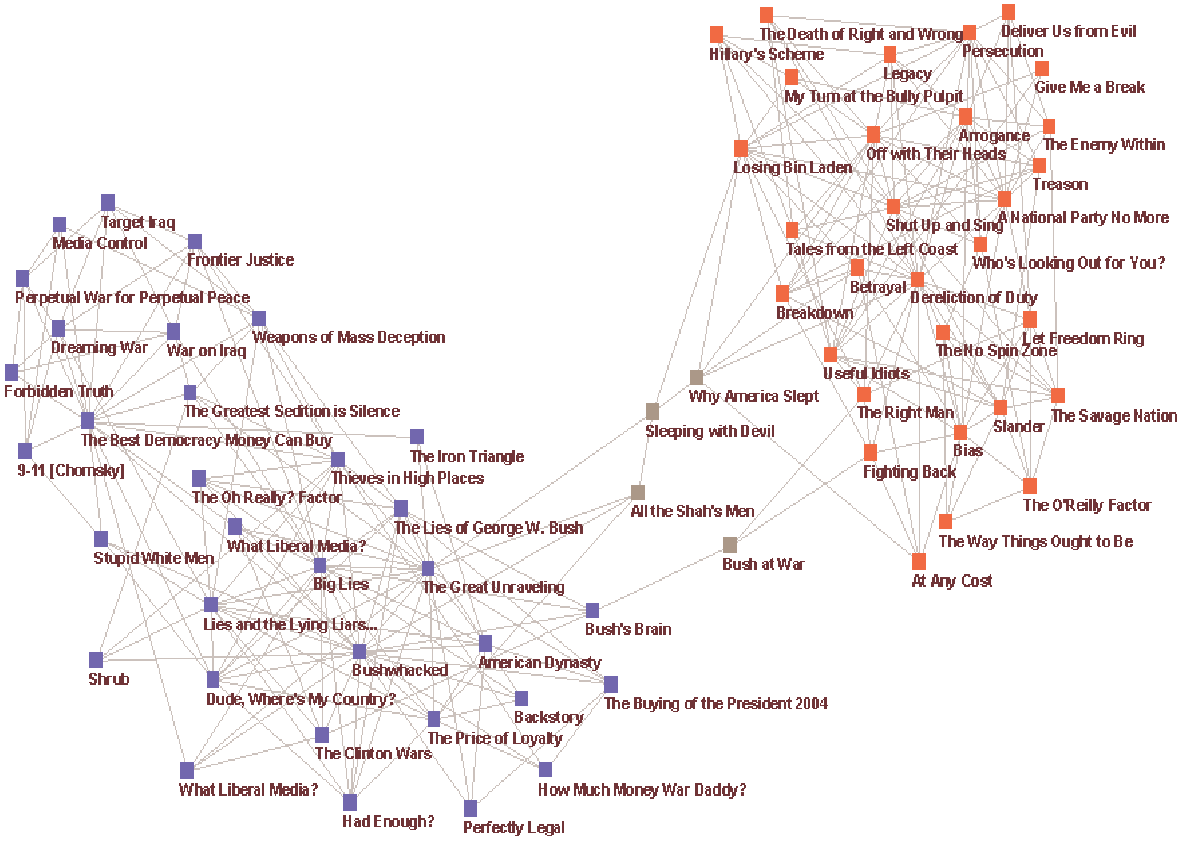
Outline

- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
    - Enforcing constraints [Ian]
    - Hierarchical [Ian]
    - Learning distances [Sugato]
    - Initializing and pre-processing [Sugato]
    - Graph-based [Sugato]

Graph-based Clustering

- Data input as graph:

  real valued edges between pairs of points denotes similarity



Constrained Graph-based Clustering

- Clustering criterion:

  minimize normalized cut

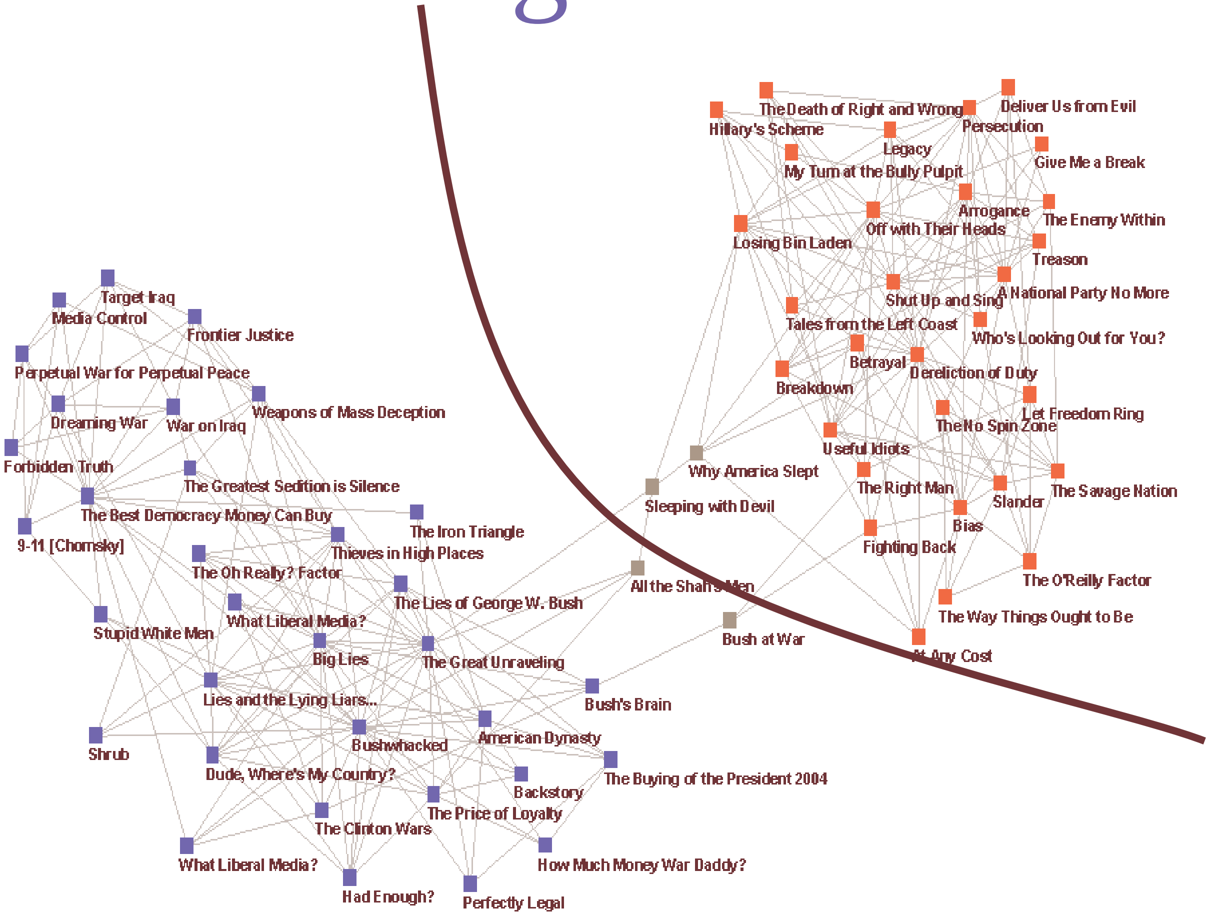
- Possible solution:

  Spectral Clustering

  [Kamvar et al. '03]

- Constrained graph clustering:

  minimize cut in input graph while maximally respecting constraints in auxilliary constraint graph

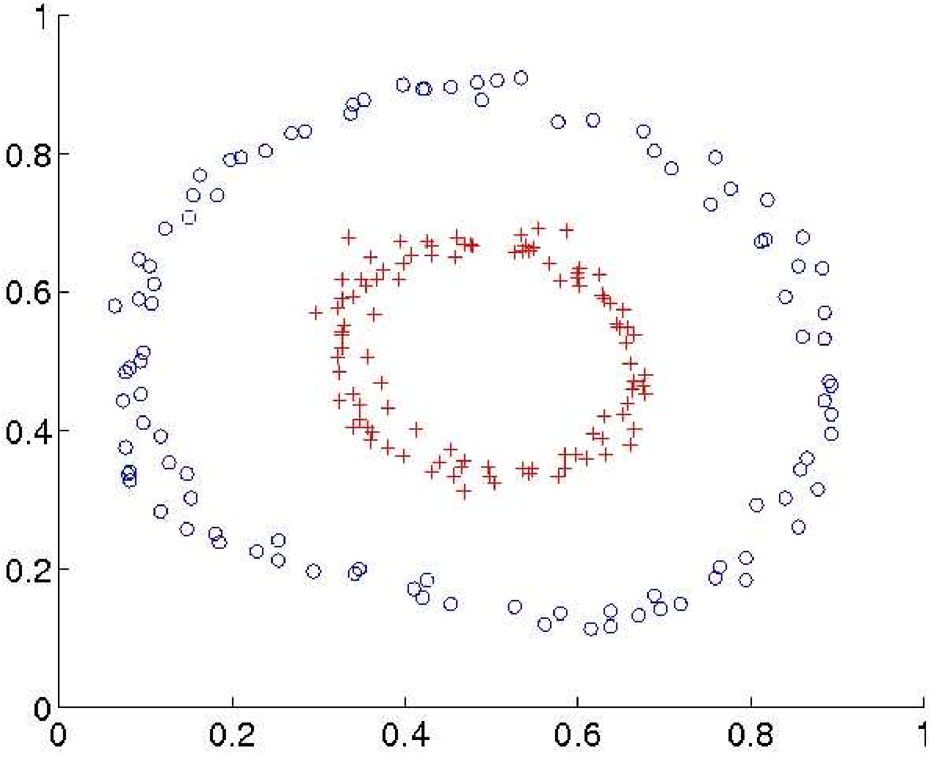


Kernel-based Clustering

- 2-circles data not linearly separable
- transform to high-D using kernel

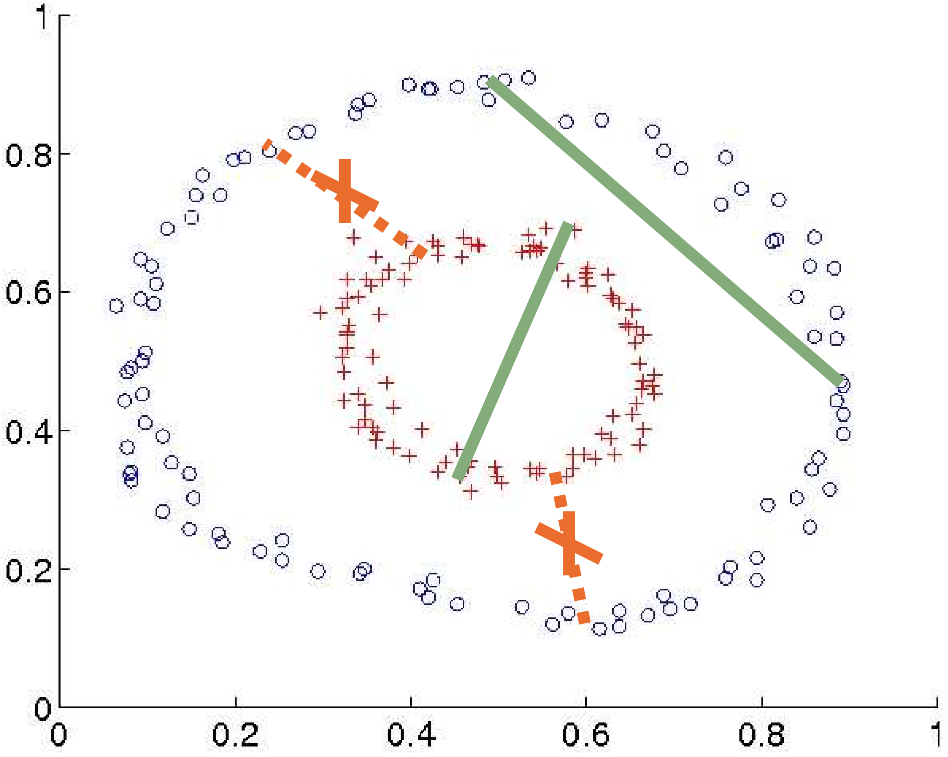
$$e.g., < s_1, s_2 >= e^{-\|s_1 - s_2\|^2}$$

- cluster kernel similarity matrix using weighted kernel K-Means



Constrained Kernel-based Clustering

- Use the data and the specified constraints to create appropriate kernel



SS-Kernel-KMeans [Kulis et al.’05]

- Contributions:
  - Theoretical equivalence between constrained graph clustering and weighted kernel KMeans
  - Uses kernels to unify vector-/graph- based constrained clustering
- Algorithm:
  - Forms a kernel matrix from data and constraints
  - Runs weighted kernel KMeans
- Benefits:
  - HMRF-KMeans and Spectral Clustering are special cases
  - Fast algorithm for constrained graph-based clustering (no spectral decomposition necessary)
  - Kernels allow constrained clustering with non-linear cluster boundaries

Kernel for HMRF-KMeans with squared Euclidean distance

$$J_{HMRF} = \sum_{c=1}^{k} \sum_{s_i \in S_c} \| s_i - C_c \|^2 - \sum_{\substack{(s_i, s_j) \in ML \\ s.t. l_i = l_j}} \frac{w_{ij}}{|S_{l_i}|} + \sum_{\substack{(s_i, s_j) \in CL \\ s.t. l_i = l_j}} \frac{w_{ij}}{|S_{l_i}|}$$

$$K = S + W,$$

$$\text{where} \begin{cases} S_{ij} = s_i . s_j, \\ W_{ij} = \begin{matrix} + w_{ij} \text{ if } (s_i, s_j) \in ML \\ - w_{ij} \text{ if } (s_i, s_j) \in CL \end{matrix} \end{cases}$$

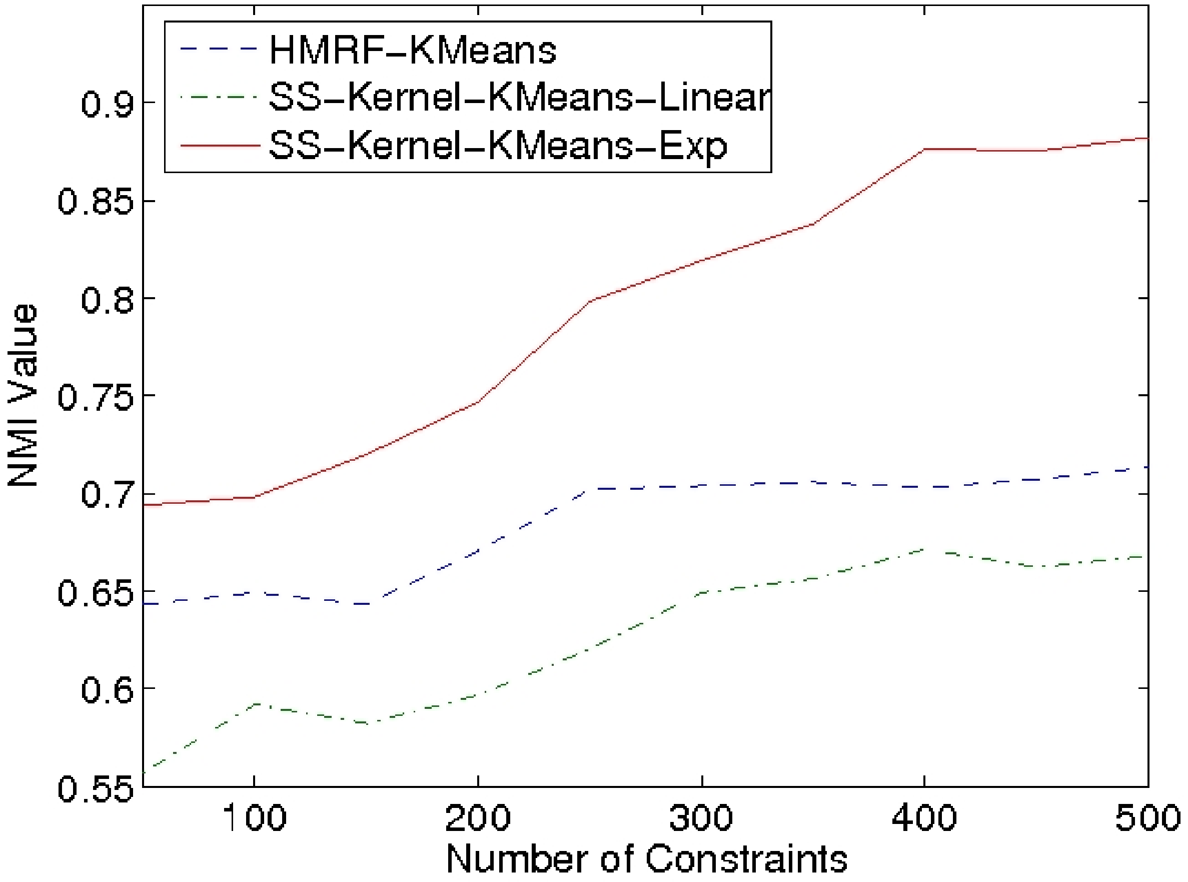
Kernel for Constrained Normalized-Cut Objective

$$J_{NormCut} = \sum_{c=1}^{k} \frac{\text{links}(V_c, V \setminus V_c)}{\deg(V_c)} - \sum_{\substack{(s_i,s_j)\in ML \\ s.t. l_i = l_j}} \frac{w_{ij}}{\deg(V_{l_i})} + \sum_{\substack{(s_i,s_j)\in CL \\ s.t. l_i = l_j}} \frac{w_{ij}}{\deg(V_{l_i})}$$

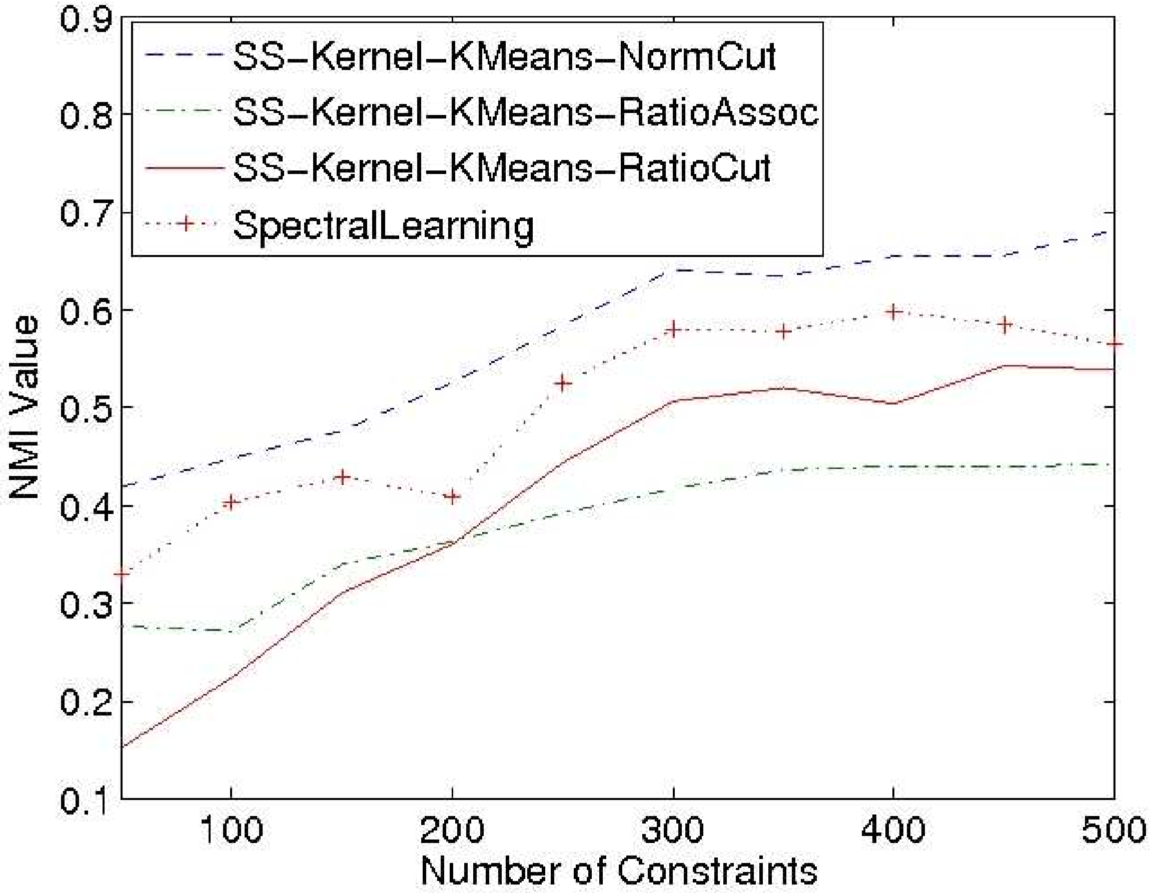
$$K = D^{-1}AD + D^{-1}WD,$$

$$\text{where} \begin{cases} A_{ij} = \text{graph affinity}\,(i,\, j), \\ D = \text{diagonal degree matrix} \\ W_{ij} = \begin{matrix} +w_{ij} \text{ if } (s_i, s_j) \in ML \\ -w_{ij} \text{ if } (s_i, s_j) \in CL \end{matrix} \end{cases}$$

Experiment: PenDigits subset



Experiment: Yeast Gene network



Today we talked about …

- Introduction and Motivation [Ian]
- Uses of constraints [Sugato]
- Real-world examples [Sugato]
- Benefits and problems of using constraints [Ian]
- Algorithms for constrained clustering
  - Enforcing constraints [Ian]
  - Hierarchical [Ian]
  - Learning distances [Sugato]
  - Initializing and pre-processing [Sugato]
  - Graph-based [Sugato]

Thanks for Your Attention. We Hope You Learnt a Few Things

Feel free to ask us questions during the conference

References – 1

1] A. Banerjee and J. Ghosh. Frequency Sensitive Competitive Learning for Balanced Clustering on High-dimensional Hyperspheres. In IEEE Transactions on Neural Networks, 2004.

[2] N. Bansal, A. Blum and S. Chawla, ”Correlation Clustering”, 43rd Symposium on Foundations of Computer Science (FOCS 2002), pages 238-247.

[3] S. Basu, A. Banerjee and R. J. Mooney, ”Semisupervised Learning by Seeding”, Proc. 19th Intl. Conf. on Machine Learning (ICML-2002), Sydney, Australia, July 2002.

[4] S. Basu, M. Bilenko and R. J. Mooney, ”A Probabilistic Framework for Semi-Supervised Clustering”, Proc. 10th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD-2004), Seattle, WA, August 2004.

[5] S. Basu, M. Bilenko and R. J. Mooney, ”Active Semi-Supervision for Pairwise Constrained Clustering”, Proc. 4th SIAM Intl. Conf. on Data Mining (SDM-2004).

[6] K. Bennett, P. Bradley and A. Demiriz, ”Constrained K-Means Clustering”, Microsoft Research Technical Report 2000-65, May 2000.

[7] De Bie T., Momma M., Cristianini N., “Efficiently Learning the Metric using Side-Information”, in Proc. of the 14th International Conference on Algorithmic Learning Theory (ALT2003), Sapporo, Japan, Lecture Notes in Artificial Intelligence, Vol. 2842, pp. 175-189, Springer, 2003.

[8] M. Bilenko, S. Basu. A Comparison of Inference Techniques for Semi-supervised Clustering with Hidden Markov Random Fields. In Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (SRL-2004), Banff, Canada, July 2004

References – 2

[9] A. Blum, J. Lafferty, M.R. Rwebangira, R. Reddy, ”Semi-supervised Learning Using Randomized Mincuts”, International Conference on Machine Learning, 2004.

[10] M. Charikar, V. Guruswami and A. Wirth, ”Clustering with Qualitative Information”, Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003.

[11] H. Chang, D.Y. Yeung. Locally linear metric adaptation for semi-supervised clustering. Proceedings of the Twenty-First International Conference on Machine Learning (ICML), pp.153-160, Banff, Alberta, Canada, 4-8 July 2004.

[12] D. Cohn, R. Caruana, and A. McCallum, ”Semi-supervised clustering with user feedback”, Technical Report TR2003-1892, Cornell University, 2003.

[13] H. Daume and D. Marcu. A Bayesian Model for Supervised Clustering with the Dirichlet Process Prior. In JMLR 6, pp. 1551–1577, 2005.

[14] I. Davidson, S.S. Ravi, Clustering under Constraints: Feasibility Results and the K-Means Algorithm, SIAM Data Mining Conference 2005.

[15] I. Davidson, S.S. Ravi, Hierarchical Clustering with Constraints: Theory and Practice, ECML/PKDD 2005.

[16] I. Davidson, S.S. Ravi, Identifying and Generating Easy Sets of Constraints For Clustering, AAAI 2006.

[17] I. Davidson, S.S. Ravi, The Complexity of Non-Hierarchical Clustering With Instance and Cluster Level Constraints, To Appear Journal of Knowledge Discovery and Data Mining.

References – 3

[18] I. Davidson, K. Wagstaff, S. Basu, Measuring Constraint-Set Utility for Partitional Clustering Algorithms, ECML/PKDD 2006.

[19] E. D. Demaine and N. Immorlica. Correlation Clustering with Partial Information. 6th Approximation Algorithms for Combinatorial Optimization Problems and 7th Randomization and Approximation Techniques in Computer Science Workshops (RANDOM-APPROX 2003)

[20] A. Demiriz, K. Bennett and M.J. Embrechts. Semi-supervised Clustering using Genetic Algorithms. In ANNIE'99 (Artificial Neural Networks in Engineering), November 1999

[21] A. S. Galanopoulos and S. C. Ahalt. Codeword distribution for frequency sensitive competitive learning with one-dimensional input data. IEEE Transactions on Neural Networks, 7(3):752-756, 1996.

[22] M. R. Garey and D. S. Johnson and H. S. Witsenhausen. The complexity of the generalized Lloyd-Max problem. IEEE Transactions on Information Theory, 28(2):255-256, 1982 j

[23] David Gondek, Shivakumar Vaithyanathan, and Ashutosh Garg Clustering with Model-level Constraints, SIAM International Conference on Data Mining (SDM), 2005.

[24] David Gondek and Thomas Hofmann Non-Redundant Data Clustering, 4th IEEE International Conference on DataMining (ICDM), 2004.

[25] T. F. Gonzalez. Clustering to Minimize the Maximum Intercluster Distance. In Theoretical Computer Science, Vol. 38, No. 2-3, June 1985, pp. 293-306.

[26] T. Hertz, A. Bar-Hillel, and D. Weinshall. Boosting margin-based distance functions for clustering. ICML 2004.

References – 4

[27] Aharon Bar Hillel. Tomer Hertz. Noam Shental. Daphna Weinshall Learning Distance Functions using Equivalence Relations ICML 2003.

[28] S. D. Kamvar, D. Klein, and C. Manning, ”Spectral Learning,” IJCAI,2003.

[29] D. Klein, S. D. Kamvar and C. D. Manning, “From Instance-Level Constraints to Space-Level Constraints: Making the Most of Prior Knowledge in Data Clustering”, Proc. 19th Intl. Conf. on Machine Learning (ICML 2002).

[30] B. Kulis, S. Basu, I. Dhillon, R. J. Mooney, ”Semi-supervised Graph Clustering: A Kernel Approach”, ICML 2005.

[31] T. Lange, M. H. C. Law, A. K. Jain and J. M. Buhmann. Learning with Constrained and Unlabelled Data. In IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[32] M. Law, Alexander Topchy, Anil K. Jain, Model-based Clustering With Probabilistic Constraints, SDM 2005.

[33] Z. Lu and T. Leen, Semi-supervised Learning with Penalized Probabilistic Clustering. NIPS 2005.

[34] N. Shental, A. Bar-Hillel, T. Hertz, and D. Weinshall, Computing Gaussian Mixture Models with EM using Side-Information. In Proc. of workshop The Continuum from labeled to unlabeled data in machine learning and data mining, ICML 2003.

[35] M. Schultz and T. Joachims, Learning a Distance Metric from Relative Comparisons, Proceedings of the Conference on Advance in Neural Information Processing Systems (NIPS), 2003.

References – 5

[36] Segal, E., Wang, H., and Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics, 19.

[37] J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. In Neural Computation, 2002 Jan;14(1):217-39.

[38] A. Strehl, J. Ghosh, R. Mooney. Impact of similarity measures on webpage clustering. AAAI Workshop on AI for Webpage Search, Austin, pp. 58-64, 2000.

[39] K. Wagstaff and C. Cardie. Clustering with Instance- Level Constraints. In Proc. 17th Intl. Conf. on Machine Learning (ICML 2000), Stanford, CA, June-July 2000, pp. 1103-1110.

[40] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl. Constrained K-means Clustering with Background Knowledge. In ICML 2001.

[41] E. Xing, A. Ng, M. Jordan, and S. Russell. Distance metric learning,with application to clustering with side-information. NIPS 15, 2003

[42] R. Yan, J. Zhang, J. Yang and A. Hauptmann A Discriminative Learning Framework with Pairwise Constraints for Video Object Classification In IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR), 2004.

[43] Z. Zhang, J.T. Kwok, D.Y. Yeung. Parametric distance metric learning with label information. Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI’03), pp.1450-1452, Acapulco, Mexico, August 2003.

[44] S. Zhong and J. Ghosh. Scalable, model-based balanced clustering. In SIAM International Conference on Data Mining (SDM-03), pp.71-82, San Francisco, CA, 2003.